



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño y Simulación del sistema de control de apogeo de
un cohete sonda. Proyecto Faraday Rocketry UPV.

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

AUTOR/A: González Chacón, Antoni

Tutor/a: García-Nieto Rodríguez, Sergio

CURSO ACADÉMICO: 2021/2022

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Trabajo de Fin de Grado

Libro 1: Memoria

**Diseño y simulación del sistema de control de apogeo de un cohete sonda.
Proyecto Faraday Rocketry UPV.**



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Autor: Antoni González Chacón

Tutor: Sergio García-Nieto Rodríguez

Grado en Ingeniería Aeroespacial

Curso 2021-2022

Resumen

Este proyecto aborda el diseño del sistema de control de apogeo encargado de accionar el movimiento de unos aerofrenos instalados en un cohete sonda desarrollado por el grupo de generación espontánea de la univestidad politécnica de Valencia, Faraday Rocketry UPV.

Dichos aerofrenos pretenden controlar la velocidad del cohete con el objetivo de llegar a un valor de apogeo deseado.

Para poder llevar a cabo esta tarea va a ser necesario modelar dinámicamente el cohete, así como sus aerofrenos. Con ese modelo, se diseñará un sistema de control para cumplir con las especificaciones establecidas y finalmente se desarrollará un entorno de simulación en Matlab Simulink donde se introducirá el modelo dinámico del cohete y los elementos del sistema de control para comprobar su funcionamiento y visualizar los resultados

Abstract

This project deals with the design of the apogee control system in charge of the movement of the airbrakes installed in a sounding rocket developed by the Generación Espontánea group from the Polytechnic University of Valencia, Faraday Rocketry UPV.

These airbrakes are intended to control the speed of the rocket in order to reach a desired apogee value.

In order to carry out this task, it will be necessary to develop the dynamic equations that defines the rocket's behavior, as well as its airbrakes'. With this model, it will be designed a control system to reach the established specifications and finally it will be developed a simulation environment in Matlab Simulink where the dynamic model of the rocket and the elements of the control system will be introduced to visualize the results and validate its operation.

Agradecimientos

A todos mis compañeros del Faraday por hacer un trabajo imposible que me permite hacer este proyecto, en especial a Nacho por compartir tema conmigo y darme trabajo hecho y a Pepe por dar la idea en un inicio.

A los profesores de mi carrera que han hecho que me interese y me encante mi carrera en especial Pedro Piqueras, Miguel Angel Mateo Pla y por supuesto a Sergio García-Nieto.

A los 3g por hacer que me encante estar en Valencia y por no apoyarme nunca a la vez que me apoyan. A na Mariperi perquè sino me renyes i a sa meva família per donarme s'oportunitat i es benestar per poder fer això.

Índice

Resumen	I
Abstract	I
Agradecimientos	II
Índice	III
Índice de figuras	V
Índice de gráficas	VII
1.1. INTRODUCCIÓN	1
1.1.1. Objeto	1
1.1.2. Justificación del Proyecto	1
1.2. ESTADO DEL ARTE	2
1.2.1. Definición	2
1.2.2. Historia de la cohetería	2
1.3. ESQUEMA GENERAL DE LOS COHETES SONDA	5
1.3.1. Sistema de propulsión	5
1.3.2. Telecomunicaciones	6
1.3.3. Aviónica	7
1.3.4. Sistema de recuperación	8
1.3.5. Estación de tierra	9
1.3.6. Carga de pago	9
1.4. PROYECTO DE FARADAY ROCKETRY UPV	9
1.4.1. Presentación	9
1.4.2. Configuración del cohete	10
1.5. DESCRIPCIÓN DEL PROYECTO	11
1.5.1. Limitaciones	11
1.5.2. Soluciones alternativas	11
1.5.3. Descripción general de la solución adoptada	13

1.6. DESCRIPCIÓN DEL MODELO MATEMÁTICO	14
1.6.1. Hipótesis iniciales	14
1.6.2. Sistemas de referencia	15
1.6.2.1. Ejes del Cuerpo	15
1.6.2.2. Ejes de Tierra	16
1.6.3. Ángulos de Euler	16
1.6.4. Ecuaciones del movimiento	17
1.6.3. Ángulos de Euler	16
1.6.4. Ecuaciones del movimiento	17
1.6.5. Definición de los parámetros del modelo	19
1.6.5.1. Modelo propulsivo	19
1.6.5.2. Expresión de la masa	20
1.6.5.3. Posición del Centro de Gravedad	22
1.6.5.4. Tensor de Inercia	22
1.6.5.5. Expresión del Drag	24
1.7. IMPLEMENTACIÓN EN MATLAB SIMULINK	25
1.7.1. Definición de los parámetros	25
1.7.2. Funciones adicionales	28
1.7.3. Ecuación de Fuerzas	29
1.7.4. Ecuación de Momentos	31
1.7.5. Esquema del control de altitud	34
1.7.6. Señal de referencia	35
1.8. SIMULACIÓN	36
1.9. CONTROL AUTOMÁTICO	38
1.9.1. Principios del Control Automático	38
1.9.1.1. Definición de FdT y polos y ceros del sistema	39
1.9.1.2. Controladores PID	40
1.9.2. Obtención de la FdT	41
1.9.3. Ajuste del PID de los estabilizadores	43
1.9.4. Ajuste del PID de control de apogeo	45
1.10. CONCLUSIONES	49
1.11. REFERENCIAS	50

Índice de figuras

Figura 1: Páginas de Artis Magnae Artilleriae pars prima	2
Figura 2: Cohete V-2 de la Segunda Guerra Mundial	3
Figura 3: Cohete Redstone	4
Figura 4: Primera etapa del cohete Falcon 9 en su plataforma de aterrizaje	5
Figura 5: Diagrama de los diferentes sistemas de propulsión	6
Figura 6: Esquema de telecomunicaciones de un cohete	6
Figura 7: Sensores de aviónica de Faraday Rocketry UPV	7
Figura 8: Cohete en fase de recuperación con paracaídas desplegados	8
Figura 9: Primer prototipo .ASPERA”	9
Figura 10: Esquema del cohete en OpenRocket	9
Figura 11: Starship prototipo SN10 aterrizaje propulsado	12
Figura 12: Estabilizador lateral de aire en punta de Starship SN10	13
Figura 13: Esquema de ASTRA con los aerofrenos desplegados	14
Figura 14: Esquema de los Ejes de Cuerpo	16
Figura 15: Esquema de los Ejes de Tierra	16
Figura 16: Gráfica de la curva de empuje	19
Figura 17: Sección inicial de la pastilla de combustible	21
Figura 18: Maqueta de ASTRA con separación por pesos	23
Figura 19: Código de implementación de las constantes	25
Figura 20: Modelo propulsivo en Simulink	25
Figura 21: Modelo del Drag en Simulink	26
Figura 22: Implementación de la posición del CDG y derivadas	26

Figura 23: Implementación de la masa y derivadas	27
Figura 24: Implementación del Tensor de Inercia	27
Figura 25: Implementación de la derivada del Tensor de Inercia	28
Figura 26: Implementación de la matriz de rotación	28
Figura 27: Implementación del producto escalar	28
Figura 28: Implementación del producto vectorial	29
Figura 29: Implementación en Simulink del Sumatorio 1	30
Figura 30: Implementación en Simulink del Sumatorio 2 e integración	31
Figura 31: Esquema de obtención de los ángulos de Euler	31
Figura 32: Esquema de separación y reunión de los elementos de w	32
Figura 33: Implementación de los momentos inerciales	32
Figura 34: Implementación de los momentos angulares	33
Figura 35: Implementación de los momentos Externos	33
Figura 36: Implementación de la ecuación de Momentos completa	34
Figura 37: Esquema del sistema de control de apogeo	35
Figura 38: Esquema del Subsistema...	35
Figura 39: Diagrama de bloques de un sistema simple	38
Figura 40: Diagrama de bloques de un sistema de lazo abierto	38
Figura 41: Diagrama de bloques de un sistema de lazo cerrado	39
Figura 42: Diagrama de un sistema con control PID	40
Figura 43: Esquema del procedimiento para el ajuste de un PID	41

Índice de gráficas

Gráfica 1: Curva de la posición del CDG frente al tiempo	22
Gráfica 2: Gráfica de I_{xx} e I_{yy} en función del tiempo	23
Gráfica 3: Gráfica 3D del Drag en función de la velocidad y alfa	24
Gráfica 4: Trayectoria sin aerofrenos	36
Gráfica 5: Altura de la trayectoria sin aerofrenos	37
Gráfica 6: Altura de la trayectoria con aerofrenos al máximo	37
Gráfica 7: Señales de salida y entrada respectivamente	42
Gráfica 8: Ajuste de las diferentes estimaciones	42
Gráfica 9: Step Response del PID de los estabilizadores	43
Gráfica 10: Ángulos de Euler	44
Gráfica 11: LDR del proceso controlado	45
Gráfica 12: Step Response del modelo controlado	46
Gráfica 13: Error de velocidad	47
Gráfica 14: Acción del aerofreno	47
Gráfica 15: Altura de la trayectoria	48



1.1. Introducción

1.1.1. Objeto

El objetivo de este documento es el diseño del sistema de control automático del apogeo de un cohete sonda. Dicho cohete se crea por el grupo de generación espontánea Faraday Rocketry UPV. Para ello se van a obtener las ecuaciones que rigen el movimiento del cohete y se va a utilizar el software Matlab Simulink para llevar a cabo el control y la simulación, así como la obtención de resultados.

Para la validación del sistema de control diseñado, se va a incluir un entorno de simulación en Matlab Simulink que incluya los elementos necesarios para el controlador, añadiéndose al modelo completo y se van a visualizar los resultados mediante gráficas generadas por el programa.

Este documento no comprende el diseño de los elementos físicos como servomotores o materiales o formas de los actuadores, simplemente el diseño del sistema de control desde el punto de vista digital.

1.1.2. Justificación del proyecto

Faraday Rocketry UPV es un equipo de Generación Espontánea (GE) de la universidad politécnica de Valencia que nace de la voluntad de dotar a la universidad de más representación y reconocimiento en el ámbito espacial y de coherencia mediante el diseño y construcción de cohetes sonda con distintos objetivos cada uno de ellos, como la presentación a concursos de coherencia como la EUROOC, el reconocimiento nacional e internacional de batir récords de altura e incluso la utilización de dichos cohetes sonda por contratantes para portar cargas de pago de su voluntad a distintas capas de la atmósfera.

Así pues, con la creación del grupo Faraday Rocketry UPV, nace la necesidad de poner retos a corto plazo para ir avanzando en la construcción de cohetes. El primer reto que se ha propuesto el equipo es el presentarse al concurso de relevancia internacional EUROOC. En dicho concurso, el objetivo es lanzar un cohete sonda a una altura determinada y efectuar la recuperación de este de manera perfecta.

Revisando las bases del concurso se detectó que el apogeo es un factor muy importante para puntuar, ya que se pide una altura determinada, concretamente 4000 metros. Al contrario de lo que se pueda pensar, el apogeo, no es que deba superar una cierta altitud, sino que debe acercarse lo más posible a la altitud propuesta, sin sobrepasarla.

De aquí nace la necesidad de diseñar un sistema de control activo que permita ajustar la altura del apogeo al que se va a llegar durante la fase de vuelo, ya que controlar el apogeo simplemente con el empuje de los motores al inicio del lanzamiento no es un método fiable para asegurar la altitud deseada.

1.2. Estado del arte

1.2.1. Definición

Un cohete sonda es un cohete pequeño equipado con una carga de pago. Una carga de pago es un dispositivo que se le añade al cohete y que representa el motivo del lanzamiento. Esto quiere decir que un cohete sonda es la herramienta que se usa para poder llevar dispositivos para realizar experimentos, mediciones u otros objetivos a diferentes capas de la atmósfera. (WIKIPEDIA, 2021)

Generalmente, los cohetes sonda se utilizan con fines científicos, como por ejemplo medir los componentes de la atmósfera a distintas alturas. Es por esto que el tiempo de vuelo no sobrepasa los 30 minutos, normalmente entre los 5 y 20 minutos de vuelo. Este tipo de cohetes no entran en órbita, sino que describen una trayectoria parabólica, aunque si pueden llegar a alturas viables para la entrada en órbita para misiones específicas como de observación astronómica por encima de la atmósfera.

1.2.2. Historia de la cohetería

Los inicios de la cohetería nacen a partir del descubrimiento de la pólvora por los alquimistas chinos taoístas, en el siglo X. A partir de este momento, se utilizó la pólvora para la celebración en ceremonias religiosas lo que dio paso a los fuegos artificiales que representan los inicios de cohetería rudimentaria. (BLOGSPOT, 2013)

Sin embargo, el motivo de su conocimiento por Europa fue su utilización en las guerras de la conquista de Rusia y Europa del este por parte del ejército mongol de Genghis Khan, que usó la tecnología china para su aplicación en armamento de fuego.

A partir de este momento, se empezó a indagar en la pirotecnia y artillería dando lugar al primer y más consolidado libro de artillería, el *Artis Magnae Artilleriae pars prima*, del investigador polaco Kazimierz Siemienowicz, el cual se convirtió en el documento de referencia durante alrededor de dos siglos para la construcción de cohetes y otros dispositivos pirotécnicos.

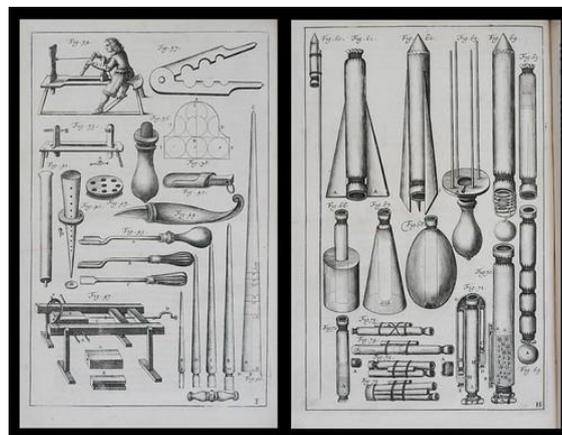


Figura 1: Páginas de *Artis Magnae Artilleriae pars prima* (LIVEJOURNAL, 2008)

Los primeros cohetes con estructura de hierro exitosos se llevaron a cabo en el siglo XVIII por parte del Reino de Mysore bajo el liderazgo del Sultán Tipu, el cual los usó como armamento en las guerras contra los británicos, los cuales se interesaron mucho por la tecnología y empezaron a investigarla y a desarrollarla masivamente durante el siglo XIX.

El momento más relevante de esta época sucede en 1844 cuando William Hale modificó la estructura y diseño de los cohetes utilizando la rotación como elemento estabilizador, cosa que mejoró los elementos anteriores de estabilización con bastones de cola largos pesados e ineficientes.

Sin embargo, los pasos más importantes de la cohetaría moderna se dieron a principios del siglo XX, cuando en 1903, Konstantin Tsiolkovski publicó *La exploración del espacio cósmico por métodos de reacción*, donde se explican los principios de los vuelos espaciales y donde se desarrollaban ecuaciones como la ecuación del cohete de Tsiolkovski, una de las ecuaciones más importantes de la propulsión de cohetes. Además, se desarrolla la utilización de oxígeno líquido como propulsor, un avance extremadamente eficiente que se sigue utilizando hoy en día.

Sin embargo, estos cohetes resultaron muy ineficientes debido a la gran pérdida de calor por los gases de escape. No fue hasta pasados los años 20 cuando Robert Goddard modificó los motores cohete con toberas supersónicas logrando duplicar el empuje conseguido hasta la época.

En esta época se destacan los científicos alemanes, que formaron la *Sociedad Alemana de Cohetes* y lograron lanzar exitosamente un cohete de propulsión líquida a alturas y distancias muy superiores a lo logrado hasta la fecha.

A partir de este momento, Alemania, aun y estar limitada por restricciones bélicas tras la primera guerra mundial, empezó a investigar exhaustivamente en el ámbito de la cohetaría, y en plena segunda guerra mundial lograron desarrollar los cohetes V-2. Estos cohetes representaban un gran avance en la cohetaría por su largo alcance y su gran capacidad de cargar pesos muy superiores, cosa que fue utilizada por los alemanes en la guerra para contener 1000 kg de amatol, un explosivo muy utilizado en la época.



Figura 2: Cohete V-2 de la Segunda Guerra Mundial (WIKIPEDIA, 2022)

Toda esa potencia balística alemana, fue aprovechada después de la guerra por los ejércitos británicos, rusos y estadounidenses, quienes compitieron por la apropiación del trabajo de los alemanes durante la guerra, y por el reclutamiento de científicos nazis.

Tras la guerra, los cohetes se usaban principalmente para cuestiones científicas, aplicándose en el lanzamiento de satélites o en pruebas científicas de telemetría y medición de parámetros atmosféricos.

Durante este período, existió la carrera espacial, que fue una rivalidad entre la Unión Soviética y Estados Unidos por el control del espacio. Esto provocó una gran cantidad de avances en la cohetaría para lograr mayores resultados con urgencia para competir con sus rivales. Durante este período, por parte de los estadounidenses, se mejoró el cohete V-2 ya mencionado creándose el Cohete Redstone, que presentaba una serie de mejoras y una mayor eficiencia. Por parte de los rusos, se lanzaron los misiles R-1, R-2 y R-5, que también se basaban en el diseño del V-2 pero se focalizaban más en el uso como misil de combate.



Figura 3: Cohete Redstone (WIKIWAND, 2022)

Por lo que respecta a las superficies de control activas, el V-2 fue el primer cohete en lograr un comportamiento de una eficiencia significativa. Se utilizaban las aletas para modificar la trayectoria por medio de efectos aerodinámicos para lograr la estabilidad deseada. (WIKIPEDIA, 2022)

Además, se han refinado los actuadores en la estabilidad del cohete, añadiéndose elementos como actuadores de aire comprimido, que se disponen de manera lateral en el cohete y generan pequeños empujes laterales que estabilizan el cohete. Además, pueden ser usados en órbita como elementos de posicionamiento para cambios de posición pequeños.

Las superficies de control y los sistemas de control de la coherencia se siguen refinando en la actualidad, habiéndose logrado por primera vez en 2013 el aterrizaje de la primera etapa del cohete Falcon 9. Este cohete se convirtió en el primero capaz de ser recuperado completamente, gracias al aterrizaje propulsado. Además, ya se está trabajando en un cohete con aterrizaje propulsado sin separación por etapas, un proyecto que ya ha logrado aterrizar el cohete Starship en pruebas de vuelo suborbitales.



Figura 4: Primera etapa del cohete Falcon 9 en su plataforma de aterrizaje (ELCONFIDENCIAL, 2016)

1.3. Esquema general de los cohetes sonda

1.3.1. Sistema de propulsión

El sistema de propulsión es el encargado de generar el empuje al cohete mediante un motor cohete. Estos motores se basan en el principio de acción-reacción para, mediante la expulsión de gases a altas velocidades, generar una fuerza en sentido contrario que conocemos como empuje. Al contrario que los motores de aviación, los motores cohete no requieren de atmósfera para funcionar, lo que hace que se tenga que llevar toda la masa que se va a expulsar dentro del cohete.

Los motores cohete funcionan expulsando los gases generados por la combustión a presiones muy elevadas de propelentes, y utilizando esa energía térmica para acelerar dichos gases a altas velocidades y ser expulsados por una tobera, un elemento que expande el gas y focaliza la dirección del chorro para obtener el empuje deseado.

Entre los combustibles más utilizados está el oxígeno líquido o el keroseno para motores cohete de combustible líquido, y el propergol sólido para motores cohete de combustible sólido. (WIKIPEDIA, 2022. *Motor cohete*)

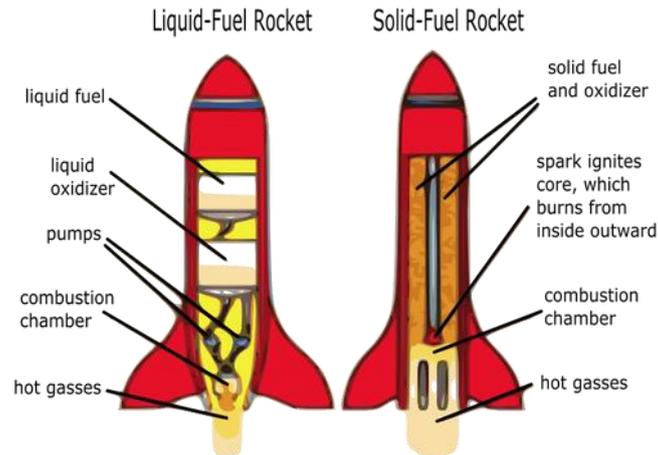


Figura 5: Diagrama de los diferentes sistemas de propulsión (PUBLICDOMAINVECTORS.ORG, 2017)

1.3.2. Telecomunicaciones

El sistema de telecomunicaciones es el encargado de mantener una comunicación activa entre el usuario y el cohete. Para establecer la comunicación se dispone de 3 canales de radio con frecuencias distintas para evitar interferencias y asegurar la comunicación. Las frecuencias de operación oscilan entre los 433 y 861 MHz.

Este sistema es el responsable de poder disponer de todos los datos medidos a tiempo real por el usuario, además de establecer la comunicación del usuario con el cohete, permitiendo la actuación sobre él, como el despliegue de paracaídas o la desintegración del cohete en caso de urgencia por orden del usuario.

Además, para modelos avanzados de cohetes, se incluye la capacidad de control remoto, permitiendo dirigir el cohete a placer por medio de este sistema de comunicación.



Figura 6: Esquema de telecomunicaciones de un cohete

1.3.3. Aviónica

El sistema de aviónica es el encargado de recopilar los datos del cohete. Se hace uso de sensores para obtener información a tiempo real e incluye todo el sistema de software de la nave, pudiendo realizar cálculos de otro tipo de parámetros que no proporcionan los sensores antes de ser enviados.

Este sistema es el encargado también de inicializar, dirigir y controlar los actuadores de la aeronave, ya sea el encendido del motor, la apertura de regiones del motor como por ejemplo el despliegue del paracaídas, o cualquier acción que esté programada para realizarse.

Consta de una placa donde se implementa un procesador con sistema operativo capaz de llevar a cabo las tareas, los sensores que van a llevar a cabo todas las mediciones necesarias para la misión y las baterías que sirven de fuente de alimentación para todos los elementos, entre otros.

Así pues, la aviónica va a ser la encargada de llevar a cabo todas las tareas de control: gestionar los datos de la aeronave, compararlos con la referencia deseada y dar la orden a las superficies de control y actuadores para su ajuste a tiempo real.

Además, al ser un cohete sonda una herramienta para transportar un experimento a alturas elevadas como se ha explicado, en caso de ser necesario alguna acción computacional en el experimento en cuestión, se podría disponer de un sistema propio del experimento para llevar a cabo estas tareas, o se podrían incluir en el sistema de aviónica del cohete sonda.



Figura 7: Sensores de aviónica de Faraday Rocketry UPV

1.3.4. Sistema de recuperación

El sistema de recuperación es el encargado de asegurar la correcta llegada de los demás sistemas a tierra, es decir, el sistema que asegura que el cohete ni ninguno de sus elementos sufre daños en el proceso de caída a tierra.

El sistema de recuperación más utilizado son los paracaídas. Estos elementos reducen drásticamente la velocidad de bajada hasta estabilizarla, es decir, reduciendo a 0 la aceleración de caída, a un valor que asegure la integridad de todos los elementos del cohete.

Generalmente se utiliza un sistema de recuperación por paracaídas en dos fases. La primera fase consiste en un paracaídas de menor medida y con gran facilidad de apertura para los primeros pasos de estabilización y para asegurar que el paracaídas principal se desplegará en la dirección correcta, ya que es común que, en caso de ser eyectado con un ángulo excesivo, se produzca el efecto cremallera. Este efecto consiste en la rotura de la estructura del cohete por el tirón de la cuerda que sujeta el paracaídas al desplegarse.

Aun así, hay otros sistemas de recuperación. Como se ha comentado anteriormente, ya se han logrado recuperaciones totales del cohete. El aterrizaje propulsado es una modalidad de sistema de recuperación ya que es el encargado de que la estructura del cohete no sufra daños.



Figura 8: Cohete en fase de recuperación con paracaídas desplegados (NTRGUADALAJARA, 2020)

1.3.5. Estación de tierra

La estación de tierra es el punto de gestión de la misión. Tiene que ser capaz de recibir los datos proporcionados por la aeronave y visualizarlos a tiempo real de manera precisa para poder tomar decisiones y actuar en caso de fallo. Desde esta estación se dan las órdenes al cohete y se visualizan todos los parámetros de la misión.

1.3.6. Carga de pago

La carga de pago es todo aquello que va dentro del cohete que no tiene una función activa en el lanzamiento. Es el motivo de la misión, lo que se quiere transportar al lugar deseado para que se realicen los objetivos de la misma.

La carga de pago la añade la parte implicada que desea hacer el experimento, pero ha de ser dimensionada, estudiada y ajustada para que se asegure la seguridad y eficiencia de su transporte.

Una carga de pago muy común son las cámaras, ya que en un gran porcentaje de los lanzamientos se añaden estos dispositivos para sacar imágenes de la tierra, del espacio, o del propio cohete en primera persona, ya sea por motivos audiovisuales o por aplicaciones científicas.

1.4. Proyecto de Faraday Rocketry UPV

1.4.1. Presentación

El cohete con el que se va a realizar el estudio es el segundo cohete en proceso de desarrollo de Faraday Rocketry UPV. El primer prototipo que se realizó, “ASPERA”, consistía en un proyecto a menor escala con el propósito de probar todos los sistemas de manera conjunta, ya que se había validado el funcionamiento de cada uno de los sistemas por separado, pero se requería de una prueba de vuelo real para verificar que las pruebas realizadas eran correctas y que todos los sistemas integrados dentro del cohete y funcionando simultáneamente funcionan correctamente.



Figura 9: Primer prototipo "ASPERA"

Una vez realizada la prueba con resultado exitoso y corroborado el funcionamiento de todos sistemas era más seguro empezar a construir el cohete objetivo del proyecto el cual es el objeto de este estudio, el cohete “ASTRA”. Consiste en un cohete sonda de combustible sólido con sistema de recuperación por paracaídas en dos etapas.

1.4.2. Configuración del cohete

En este apartado se describirán cada una de las partes del cohete ASTRA por separado aportando una pequeña explicación de sus factores más importantes

- ❖ **Fuselaje:** El fuselaje es un cilindro de 92 mm de diámetro y 2.228 m de alto hecho de material compuesto de fibra de vidrio. Consta de dos partes desmontables.
- ❖ **Ojiva:** La ojiva es la parte superior del cohete, tiene forma cónica y es extraíble fácilmente. Esto es porque el sistema de recuperación requiere del desacople de la ojiva al fuselaje para su liberación. Consta de una anilla en su parte baja que va dentro del fuselaje y donde se atan las cuerdas necesarias para efectuar su recuperación sin entorpecer la salida del paracaídas.
- ❖ **Cápsula de recuperación:** La cápsula de recuperación almacena los dos paracaídas que se van a utilizar correctamente plegados para que se facilite la apertura de los mismos. Contiene cargas de material explosivo que se accionan por medio del sistema de aviónica que son las encargadas de separar la ojiva del fuselaje y abrir correctamente los paracaídas asegurando su correcto inflado.
- ❖ **Cápsula de aviónica:** Este elemento contiene tanto el sistema de aviónica como el sistema de telecomunicaciones. Consta de dos placas base (una para cada sistema) donde se incrustan los diferentes elementos necesarios para el funcionamiento de los sistemas, incluyendo los sensores en el sistema de aviónica y elementos como la antena en el sistema de telecomunicaciones. En la parte inferior de esta cápsula es donde se pretende construir el sistema de actuadores regulados por el sistema de control que se va a explicar y diseñar en los siguientes apartados de este documento.
- ❖ **Carga de pago:** La parte destinada a la carga de pago se encuentra entre la planta propulsiva y la cápsula de aviónica. Esto es así ya que por la separación del fuselaje se hace muy accesible esta parte del cohete para realizar modificaciones y agilizar el proceso de montaje de la carga de pago.
- ❖ **Planta propulsiva:** Se trata de un motor de combustible sólido comercial desarrollado por la empresa Cesaroni. Concretamente se trata del motor “Cesaroni 4263L1350-P”.
- ❖ **Estabilizadores:** Se trata de 4 aletas dispuestas en forma de cruz simétrica que se encargan de la estabilización del cohete contrarrestando momentos minoritarios por medio de sus efectos aerodinámicos.

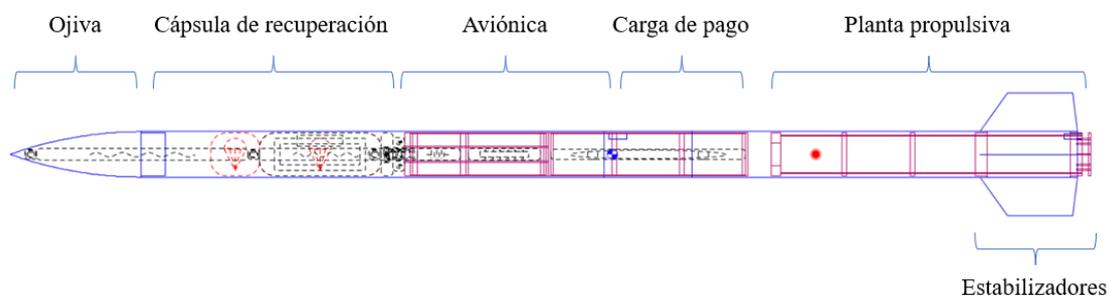


Figura 10: Esquema del cohete en OpenRocket

1.5. Descripción del proyecto

1.5.1. Limitaciones

Este estudio se ha realizado para su aplicación dentro del grupo de Generación Espontánea, sin embargo, se ha realizado dentro del departamento de I+D+I, un departamento que se encarga del diseño y estudio de mejoras del cohete en el largo plazo.

Es por eso por lo que, a la fecha de realización de este documento, no se dispone mucha de la información necesaria para su desarrollo.

De este modo, se han tomado una serie de consideraciones para aproximar lo más posible el modelo de estudio al resultado final que se desarrollará en el largo plazo por el equipo. Dichas consideraciones se explicarán a lo largo de este documento.

Además, algunos datos tomados en cuenta en el desarrollo del modelo han sido proporcionados por un miembro del equipo que ha realizado el estudio de los mismos de manera teórica, por lo que son una aproximación de la realidad, ya que no se han lanzado simulaciones precisas o medidas en túnel de viento para obtenerlos.

1.5.2. Soluciones alternativas

Como sabemos, el objetivo del trabajo es controlar el apogeo de la trayectoria del cohete para llegar a la altitud de 4000 metros.

La primera opción posible sería el diseño riguroso y preciso del sistema de propulsión del cohete. De esa manera se podría predecir el apogeo dependiendo del empuje al inicio del lanzamiento. Esta opción se basa la suposición de que las simulaciones por ordenador y el resultado real se parecerían lo suficiente como para no tener que disponer de un sistema activo para regularlo.

Sin embargo, esta solución presenta una serie de problemas. Primeramente, existe una gran dificultad en diseñar un motor cohete sólido con unas especificaciones tan concretas, ya que con el mismo motor cohete se pueden obtener dos curvas de empuje ligeramente diferentes, lo cual no presenta una gran diferencia, pero en términos de rigurosidad es una solución bastante imprecisa. Además, no disponer de un sistema activo hace que se tenga que lanzar el cohete de una manera concreta para cada situación atmosférica dependiendo de la densidad del aire, el viento y otra serie de factores que afectan a la trayectoria del cohete, lo cual es inviable al no conocer de antemano esa situación atmosférica con tanta precisión.

Por todo esto es por lo que se ha visto la necesidad de diseñar un sistema de control activo. Así pues, hay tres maneras de controlar el apogeo:

- Diseñar un sistema activo que acelere el cohete en caso de no llegar a la altura deseada con el empuje inicial,
- Diseñar un sistema activo que frene el cohete en caso de sobrepasarla, o
- Disponer de varios sistemas para realizar ambas tareas simultáneamente para asegurar el control en las dos situaciones.

Para cada uno de estos escenarios se barajaron distintas soluciones:

Para el primer caso, se consideró el uso de la misma propulsión del cohete como acelerador. De este modo se diseñaría un sistema de control que encendiera el motor en caso de necesitarlo en mitad del vuelo. Sin embargo, un cohete de motor sólido no se puede controlar su apagado ni se puede reencender por la misma naturaleza del motor, por lo que sería necesaria la creación de un motor líquido que sí permite el apagado y reencendido del mismo a voluntad. Así, en caso de no llegar a la altura deseada, se reencendería el motor para dar más cantidad de empuje durante el vuelo de ascenso.

Esta opción presenta una serie de ventajas, ya que un motor líquido y sus características de reencendido presentan una gran cantidad de aplicaciones más allá del control de apogeo. Se ha barajado esta opción para poder realizar un aterrizaje propulsado, así, no sería necesario el uso de paracaídas como sistema de recuperación y se aseguraría la integridad estructural del cohete en la fase de aterrizaje.



Figura 11: Starship prototipo SN10 aterrizaje propulsado (VIDEOFROMSPACE, 2021)

Sin embargo, aunque presente posibilidades en otros campos de la misión, en el tema que ocupa a este documento que es el control de apogeo no es viable su utilización. Primeramente, un reencendido presenta un empuje muy agresivo y difícil de controlar, tanto la dirección como la cantidad de empuje en el primer instante del encendido, lo que podría resultar en una desestabilización completa del cohete o en la aplicación de un valor excesivo de empuje que hiciera que se superara con creces la altura deseada. Además, cabe destacar que la creación de un motor cohete líquido es muy costosa y fuera del alcance de Faraday Rocketry UPV por el momento, y más si el motivo principal de su creación es el control automático del apogeo. Por todo esto, esta opción resulta inviable.

Siguiendo con la idea de utilizar el empuje como actuador y dado que un motor líquido es inviable para el control, se pueden utilizar actuadores de aire comprimido para realizar el control. Consiste en unas válvulas que expulsan aire comprimido a pequeñas cantidades.

Esta opción presenta grandes ventajas en el control ya que estos actuadores generalmente se utilizan colocándose en los laterales de la nave para estabilizar el cohete, por lo que, si se realiza la labor de modelaje de los mismos para el control de apogeo, se puede utilizar ese modelo con pequeñas modificaciones para generar empuje en cualquier dirección en la que se ubiquen los actuadores y utilizarlos también como estabilizadores. Así pues, se puede considerar el ubicarlos en la parte inferior de esta para generar un pequeño empuje controlado y acelerar el cohete en el primero de los escenarios mencionados.



Figura 12: Estabilizador lateral de aire en punta de Starship SN10 (VIDEOFROMSPACE, 2021)

Esta solución además permite el mejor escenario que es controlar el apogeo tanto frenando el cohete como acelerándolo, dependiendo de la localización de los actuadores. Es decir, en caso de disponer de dos juegos de actuadores, unos ubicados en la parte baja y otros ubicados en la ojiva, se puede controlar el apogeo para los dos escenarios mencionados anteriormente.

Sin embargo, el estudio propulsivo de estos actuadores es una labor muy costosa, además de complicar mucho la construcción y montaje dentro del cohete, ya que se necesita mucha precisión en la dirección de estos para no generar momentos desestabilizantes. Además, con la disposición del cohete actualmente sería inviable, ya que se debería poner uno de estos actuadores en la ojiva del cohete, donde actualmente se encuentra el sistema de recuperación, y otro en la parte inferior, donde actualmente el motor ocupa todo el espacio disponible, por lo que se complicaría mucho la logística.

1.5.3. Descripción general de la solución adoptada

Así pues, la solución adoptada se ha decidido que sean unos aerofrenos localizados en el fuselaje, en la parte inferior de la cápsula de aviónica que se sitúa en el centro del cohete. Se trata de 4 placas de aluminio dispuestas en forma de X simétrica sobre las que actúa un servomotor para controlar su porcentaje de salida.

El principio de funcionamiento de los mismos se basa en la modificación controlada de la cantidad de resistencia aerodinámica del cohete (también llamada “Drag”). Si bien el cohete en si ya tiene una cierta resistencia aerodinámica, lo que se consigue con las placas de aluminio es aumentar esa resistencia aerodinámica en la dirección tangente a la trayectoria. Cuanta más drag, menos velocidad tiene el cohete en esa dirección, y por tanto menor altura consigue lograr el cohete.

Esta solución facilita el modelaje ya que solo se tiene en cuenta el drag como parámetro variable en función del porcentaje de salida de los aerofrenos, además de que su montaje dentro del cohete se simplifica mucho, puesto que solo se necesita de un servomotor y unos raíles para controlar el movimiento de los aerofrenos.

Además, y muy importante, la disposición de los aerofrenos deja la puerta abierta para realizar en un futuro un control de la actitud del cohete. Esto es porque si se consigue controlar independientemente cada uno de los 4 aerofrenos, se pueden desplegar diferentes porcentajes de salida cada uno de ellos. Esto generaría diferentes valores de Drag en cada uno de los lados del cohete. A diferencia que en el caso de tener el mismo porcentaje de salida en los 4 aerofrenos, que el drag de cada uno de ellos se contrarresta con el aerofreno que tienen delante ya que los 4 generan el mismo valor de Drag, en caso de salir con diferente porcentaje generaría más Drag de un lado que de otro, generando un momento en el cohete que resultaría en una variación de la actitud. Controlando esto con precisión, se puede llegar a dirigir el cohete automáticamente o incluso remotamente por el usuario.

Así pues, controlando el porcentaje de salida de estos aerofrenos, se modifica el drag del cohete y se reduce la velocidad, y a consecuencia el apogeo al que va a llegar el cohete. Cabe destacar que este modelo parte de la idea de que el apogeo al que se llegaría con el empuje proporcionado sin la actuación de los aerofrenos es mayor al deseado, por lo que para ser efectivo este modelo, se tiene que diseñar la trayectoria con un valor de empuje ligeramente superior al necesario para llegar al apogeo deseado, para que pueda ser frenado controladamente y precisar la altura máxima a la que se llega.



Figura 13: Esquema de ASTRA con los aerofrenos desplegados

1.6. Descripción del modelo matemático

1.6.1. Hipótesis iniciales

El vuelo del cohete presenta tres partes principales: el movimiento a través del raíl, que estabiliza y dirige el cohete en los primeros instantes del lanzamiento, el movimiento libre hasta el apogeo y el trayecto de caída con paracaídas, que se puede desglosar en dos ya que se despliegan dos paracaídas con distinto comportamiento. Sin embargo, en este trabajo se van a asumir dos cosas: que no existe raíl, es decir, que se

lanza desde el suelo sin plataforma de direccionamiento, y que la tercera fase es irrelevante, ya que el control se realiza únicamente en la fase de ascenso y a partir del apagado del motor, ya que no tiene sentido desplegar los aerofrenos en la fase de caída. A partir del apogeo, los aerofrenos se recogerán y no es objeto de este proyecto modelar el comportamiento del cohete a partir de ese momento.

Así pues, en la fase de vuelo que nos interesa, se considera que el cohete dispone de 6 grados de libertad (o 6 DoF haciendo referencia a sus siglas en inglés), 3 de rotación y 3 de traslación. En términos de velocidad, se corresponden respectivamente a los parámetros u , v , w para la velocidad lineal en los tres ejes, correspondientes al movimiento de traslación, y p , q , r para la velocidad angular en los tres ejes correspondientes al movimiento de rotación. Dichos ejes son los ejes del body-frame que se explicarán en el siguiente apartado.

Para la obtención de las ecuaciones, se consideran las siguientes hipótesis:

- ❖ El cohete es un cuerpo rígido, es decir, dos puntos cualesquiera del cohete mantendrán su distancia durante todo el vuelo.
- ❖ Se considera despreciable el giro de la Tierra por lo que no se aplicarán términos del efecto Coriolis.
- ❖ Se asume viento nulo durante todo el proceso.
- ❖ Se asume que las características de la atmósfera son constantes durante todo el vuelo, por lo que el Drag no va a depender de la altura sino tan solo de la velocidad del cohete.
- ❖ Se asume un estado de atmósfera en calma, sin turbulencias.
- ❖ Se asume una construcción ideal de los elementos físicos del cohete por lo que se establecen dos planos de simetría. ($I_{XY} = I_{XZ} = I_{YZ} = 0$)
- ❖ La masa en vacío del cohete permanece constante durante todo el vuelo, sin considerar desprendimiento de residuos de combustible tras el apagado del moto, roturas de la estructura o liberación de la carga de pago en su caso.

En caso de considerar una misión que incumpla cualquiera de estas hipótesis se deberán modificar las ecuaciones del movimiento desarrolladas en los apartados siguientes.

1.6.2. Sistemas de referencia

Este proyecto contempla dos sistemas de referencia, ambos orientados por la regla de la mano derecha ortogonal: Ejes del Cuerpo y Ejes de Tierra.

1.6.2.1. Ejes del Cuerpo (^B)

Este sistema de referencia (también llamado body-frame) representa el cohete. Sus ejes se mantienen inmóviles respecto a la orientación y posición del cohete y está situado con origen en el centro de masa en vacío del mismo. Sus ejes se definen como:

- \widehat{b}_1 : Eje perpendicular a la superficie del fuselaje
- \widehat{b}_2 : Eje perpendicular a la superficie del fuselaje y a \widehat{b}_1
- \widehat{b}_3 : Eje perpendicular a \widehat{b}_1 y \widehat{b}_2 , se corresponde con la dirección tangente al fuselaje

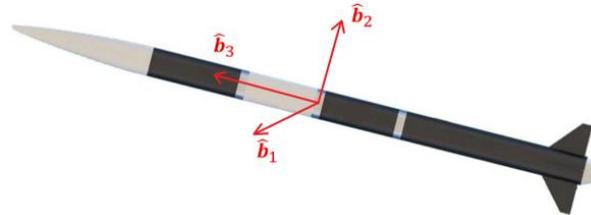


Figura 14: Esquema de los Ejes de Cuerpo

1.6.2.2. Ejes de Tierra (A)

Este sistema de referencia está fijo y anclado a un punto concreto de la tierra, generalmente en la estación de tierra. Sus ejes forman también un triedro ortogonal y se definen de la siguiente manera:

- \widehat{a}_1 y \widehat{a}_2 : Perpendiculares entre sí y tangentes a la superficie de la Tierra.
- \widehat{a}_3 : Eje perpendicular a \widehat{b}_1 y \widehat{b}_2 , se corresponde con la dirección perpendicular a la superficie de la tierra y hacia arriba



Figura 15: Esquema de los Ejes de Tierra

1.6.3. Ángulos de Euler

Los ángulos de Euler son 3 elementos que expresan la actitud del cohete en relación a un sistema de referencia externo como puede ser el Eje de Tierra. Estos ángulos son los siguientes:

- Roll (Φ): Se corresponde con el ángulo de rotación respecto al eje \widehat{b}_1
- Pitch (θ): Representa el ángulo de rotación respecto al eje \widehat{b}_2
- Yaw (ψ): Se corresponde con el ángulo de rotación respecto al eje \widehat{b}_3

Estos ángulos son muy útiles para conocer en todo momento la actitud del cohete, nos marcan la estabilidad del mismo y se pueden utilizar en el cálculo para cambiar de sistema de referencia en Tierra a body-frame. Para lograrlo, se hace uso de la matriz de transformación:

Para calcular la matriz de transformación se hace uso de la matriz de cosenos, que permite cambiar de sistema de referencia según el siguiente esquema:

$$C_B^A = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Donde c_{ij} es el coseno del ángulo entre el eje i del marco de referencia y el eje j del cuerpo. Traduciendo a ángulos de Euler, surgen las siguientes expresiones:

$$\Phi = \arctan\left(\frac{c_{32}}{c_{33}}\right) \quad \theta = \arctan(-c_{31}) \quad \psi = \arctan\left(\frac{c_{21}}{c_{11}}\right)$$

Así pues, realizando una serie de sustituciones y ajustes, la matriz de transformación en función de los ángulos de Euler queda de la siguiente manera:

$$R_B^A = \begin{pmatrix} \cos(\theta)\cos(\psi) & -\cos(\Phi)\sin(\psi) & \sin(\Phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & +\sin(\Phi)\sin(\theta)\sin(\psi) & +\cos(\Phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & +\sin(\Phi)\cos(\theta) & \cos(\Phi)\cos(\theta) \end{pmatrix} \quad (1.1)$$

Que permite realizar cambios del sistema de referencia del tipo: $\mathbf{r}^A = R_B^A \cdot \mathbf{r}^B$, siendo \mathbf{r} cualquier vector de tres elementos.

1.6.4. Ecuaciones del movimiento

Para obtener las ecuaciones dinámicas del movimiento del cohete se necesitan 6 ecuaciones, 3 de traslación y 3 de rotación, es decir, 3 de Fuerzas y 3 de Momentos. Así pues, se desarrollarán 2 ecuaciones matriciales, una reuniendo los términos que afectan al equilibrio de Fuerzas, y en otra los que afectan al equilibrio de Momentos.

Empezando por el equilibrio de Fuerzas se deben tener en cuenta 3 partes principales: la aceleración lineal en los 3 ejes del cohete, los efectos de variación de masa, y las fuerzas externas que actúan en el cohete.

$$m \cdot \mathbf{a} = \mathbf{F}_{masa_variable} + \mathbf{F}_{externas} \quad (1.2)$$

Hay 5 términos que regulan el comportamiento de la aceleración lineal: la derivada de la velocidad lineal, la aceleración lineal inducida por la aceleración angular de la aeronave a causa del movimiento del centro de gravedad, la rotación de la nave a causa de la velocidad lineal inducida por movimiento del centro de gravedad, la

aceleración del centro de gravedad, y la aceleración inducida de la velocidad del centro de gravedad. Así pues, la aceleración queda de la siguiente forma:

$$\mathbf{a} = \dot{\mathbf{v}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}_{CM} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{CM}) + \mathbf{r}_{CM}'' + 2\boldsymbol{\omega} \times \mathbf{r}_{CM}' \quad (1.3)$$

En cuanto a los efectos de la masa variable, influyen el empuje y los términos dependientes de las derivadas de la ecuación de la masa respecto al tiempo:

$$\mathbf{F}_{masa_variable} = \mathbf{T} - 2\dot{m}\mathbf{r}_{CM}' + 2\boldsymbol{\omega} \times (\mathbf{r}_{noz} - \mathbf{r}_{CM}) + \ddot{m}(\mathbf{r}_{noz} - \mathbf{r}_{CM}) \quad (1.4)$$

Finalmente, simplemente se tienen que considerar dos factores externos que influyen en el movimiento del cohete, la gravedad y la resistencia aerodinámica.

$$\mathbf{F}_{externas} = -\mathbf{D} - R_A^B (mg\hat{\mathbf{a}}_3) \quad (1.5)$$

De esta manera, la ecuación no lineal de equilibrio de fuerzas del movimiento del cohete queda de la siguiente manera:

$$m(\dot{\mathbf{v}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}_{CM} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{CM}) + \mathbf{r}_{CM}'' + 2\boldsymbol{\omega} \times \mathbf{r}_{CM}') = \mathbf{T} - 2\dot{m}\mathbf{r}_{CM}' + 2\boldsymbol{\omega} \times (\mathbf{r}_{noz} - \mathbf{r}_{CM}) + \ddot{m}(\mathbf{r}_{noz} - \mathbf{r}_{CM}) - \mathbf{D} - R_A^B (mg\hat{\mathbf{a}}_3) \quad (1.6)$$

Por lo que respecta al equilibrio de momentos, se deben tener en cuenta los términos que influyen en el momento generado por fuerzas externas, el momento generado por la variación de la masa por medio del tensor de inercia, y los momentos inerciales.

$$\mathbf{M} + \mathbf{M}_{inerciales} = \mathbf{M}_{externos} \quad (1.7)$$

En primer lugar, la variación de la masa hace que se tenga que tener en cuenta no solo el tensor de inercia sino también su derivada respecto al tiempo, los términos implicados son los siguientes:

$$\mathbf{M} = \mathbf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega}) + \mathbf{I}' \cdot \boldsymbol{\omega} \quad (1.8)$$

Además, se tienen que considerar los momentos inerciales, cuya expresión incluye únicamente el término inercial de la aceleración lineal.

$$\mathbf{M}_{inerciales} = m\mathbf{r}_{CM} \times \dot{\mathbf{v}} \quad (1.9)$$

Por último, se tienen que incluir los momentos generados por las fuerzas externas. La única fuerza externa que generará momentos es la gravedad, ya que el drag es una fuerza que siempre actúa de forma tangencial a la trayectoria, y al haber considerado los dos planos de simetría, no se generan momentos. La expresión entonces sería la siguiente:

$$\mathbf{M}_{externos} = -\mathbf{r}_{CM} \times mg\hat{\mathbf{a}}_3 + \mathbf{M}_{est} \quad (1.10)$$

Donde la variable \mathbf{M}_{est} es el momento compensatorio generado por los estabilizadores del cohete, que, en lugar de ser modelado, se va a controlar activamente mediante controladores PID en el apartado de control de este documento.

Así pues, la ecuación de equilibrio de momentos en forma matricial quedaría de la siguiente manera:

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega}) + \mathbf{I}' \cdot \boldsymbol{\omega} + m\mathbf{r}_{CM} \times \dot{\mathbf{v}} = -\mathbf{r}_{CM} \times mg\hat{\mathbf{a}}_3 + \mathbf{M}_{est} \quad (1.11)$$

1.6.5. Definición de los parámetros del modelo

Para poder realizar la simulación del vuelo del cohete utilizando el modelo explicado anteriormente, se debe conocer de antemano el valor de los parámetros del modelo.

Primeramente, hay que separar las variables anteriores en variables de estado y parámetros conocidos. Las variables de estado son las variables que describen el comportamiento del cohete en cada instante de tiempo, es decir, para un tiempo determinado t , las variables de estado nos dan la información necesaria sobre cómo se encuentra el cohete en ese momento. En este caso, las variables de estado son $[\mathbf{r} \ \mathbf{v} \ \mathbf{q} \ \mathbf{w}]$, donde \mathbf{r} es la posición del cohete respecto al sistema de referencia en Tierra, \mathbf{v} es la derivada de \mathbf{r} en función del tiempo y describe la velocidad lineal del cohete, \mathbf{q} es el vector de los ángulos de Euler, que determina la orientación del cohete, y \mathbf{w} es la derivada de \mathbf{q} en función del tiempo y se corresponde con el vector velocidad angular. Así pues, se puede conocer el estado del cohete en cada instante de tiempo mediante estas 4 variables.

Así pues, el resto de parámetros son los que condicionarán el estado del cohete, por lo tanto, tienen que ser definidos de antemano.

1.6.5.1. Modelo propulsivo

El primer parámetro que se va a definir es el empuje (\mathbf{T}). La planta propulsora del cohete es un motor comercial de combustible sólido, el “Cesaroni 4263L1350-P”. en su ficha técnica (THRUSTCURVE 2022) obtenemos datos que nos son necesarios para conocer el comportamiento de la propulsión del cohete.

Primeramente, necesitamos el tiempo de quemado, en este caso 3.2 segundos. Este parámetro nos servirá para definir el tiempo de encendido, para así organizar la ecuación de \mathbf{T} como una función a tramos, donde a partir de ese momento, la función vale 0.

Para saber el valor del empuje durante todo el tiempo que está encendido se hace uso de la curva de empuje proporcionada por la ficha técnica. Esta curva nos proporciona datos de cómo varía el empuje a través del tiempo, ya que este valor no será constante durante todo el encendido. La curva proporcionada es la siguiente:

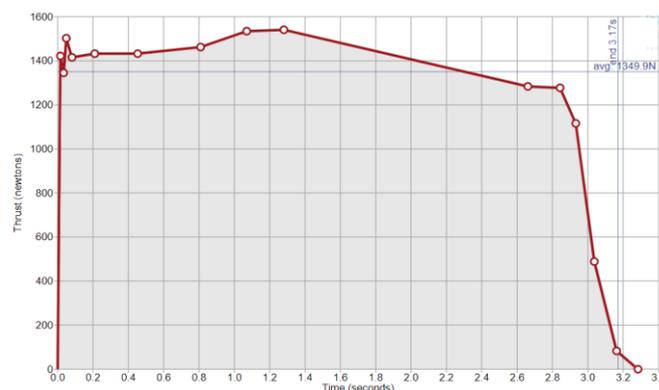


Figura 16: Gráfica de la curva de empuje (THRUSTCURVE 2022)

Como se puede ver, se nos proporcionan pocos puntos de datos en la gráfica, por lo que realizar una interpolación de la gráfica no sería apropiado. En su lugar, se ha aproximado la gráfica con dos funciones, asumiendo que mantiene un valor constante hasta el instante $t = 2.85$ s, y luego baja hasta el valor 0 de manera lineal. Esta aproximación no es muy acertada ya que se debería poner una ecuación completa para modelar bien la curva de empuje, sin embargo, es suficiente para los objetivos de este trabajo. Así pues, la ecuación del empuje queda de la siguiente manera.

$$|\mathbf{T}|(t) \begin{cases} 1425 & 0 < t < 2,85 \\ 11528 - 3591 \cdot t & 2,85 < t < 3,2 \\ 0 & 3,2 < t \end{cases} \quad (1.12)$$

Cabe destacar que esta ecuación corresponde al módulo del empuje, sin tener en cuenta la dirección en la que se aplica. Al estar en el body-frame, si se considerara que el montaje del cohete y la alineación del motor (y por lo tanto la dirección del chorro de gases de escape) son perfectos, este valor de empuje se aplicaría únicamente en el eje \widehat{b}_3 . Sin embargo, en pro de anticipar un error humano en el montaje del cohete, o simplemente de añadir un poco de imperfección al modelo para obtener unos resultados más semejantes a la realidad, se va a asumir que el empuje no se aplica perfectamente alineado con el fuselaje del cohete, sino que se considera que la dirección del chorro tiene una pequeña desviación de 1° .

Para introducir esa inclinación se va a hacer uso de la fórmula de cambio de coordenadas esféricas a cartesianas. En coordenadas esféricas, se consideraría que el ángulo α es la inclinación en el eje vertical, y el ángulo β es la dirección en el plano horizontal, mientras que $|\mathbf{T}|$ es el módulo del empuje, de esta manera, en el body-frame:

$$\mathbf{T} = |\mathbf{T}| \cdot \begin{pmatrix} \sin(\alpha) \cdot \cos(\beta) \\ \sin(\alpha) \cdot \sin(\beta) \\ \cos(\alpha) \end{pmatrix}$$

Así, introduciendo una inclinación vertical de $\alpha = 1^\circ$ y una dirección de $\beta = 45^\circ$ (para que la inclinación se realice en las dos direcciones del plano horizontal), se obtiene el vector de \mathbf{T} en el body-frame con una pequeña imperfección.

1.6.5.2. Expresión de la masa

Como se ha mencionado, el cohete se considera con masa variable, ya que se tiene en cuenta la quema del combustible, que se desprende del cohete por la tobera reduciendo el peso del mismo. Así pues, la función de la masa será la suma de la masa en vacío del cohete, que como se ha explicado se considera constante, y de la masa de combustible que hay en cada momento.

$$m(t) \begin{cases} M_c + M_{\text{fuel}}(t) & 0 < t < 3,2 \\ M_c & 3,2 < t \end{cases} \quad (1.13)$$

Donde M_c es la masa en vacío del cohete (9.969 Kg) y $M_{fuel}(t)$ es la masa de combustible en el instante de tiempo t .

Entonces, para calcular la expresión que rige la variación de la masa tenemos que calcular $M_{fuel}(t)$, y para ello tenemos que entender el proceso de quema del combustible. El motor utilizado es un motor de combustible sólido cuyas pastillas de combustible tienen forma de cilindro hueco. Esas pastillas se encienden desde la superficie interna del cilindro y se queman aumentando el radio interno del cilindro hasta quemar la pastilla por completo, es decir, hasta que el radio interno y externo son iguales.

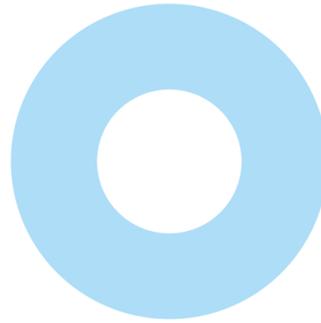


Figura 17: Sección inicial de la pastilla de combustible

Para facilitar el cálculo, se asume que el avance del quemado se produce de forma lineal, es decir, que el radio interno crece linealmente hasta igualarse al radio externo.

Así pues, la función tiene la forma del volumen del cilindro multiplicado por la densidad del combustible:

$$M_{fuel}(t) = \pi \cdot h \cdot \rho \cdot (r_{ext}^2 - r_{int}(t)^2) \quad (1.14)$$

Donde h es la altura de la pastilla de combustible, ρ es la densidad del combustible, r_{ext} es el radio exterior y $r_{int}(t)$ es la función del radio interior respecto al tiempo. Como se ha considerado avance lineal del radio interno de la pastilla de combustible, se desarrolla la función lineal del radio con valor inicial $r_{int_0} = 1.5 \text{ cm}$ y valor final $r_{int_f} = r_{ext} = 3.25 \text{ cm}$, con una pendiente de $\frac{r_{ext} - r_{int_0}}{t_f - t_0}$. Así pues, la expresión del radio interno queda de la siguiente manera:

$$r_{int}(t) = 1.5 + \frac{1.75}{3.2} \cdot t \quad (cm)$$

Que, añadiéndolo a la ecuación 1.14 y sustituyendo los valores de las variables obtenemos la siguiente ecuación de la masa de combustible.

$$M_{fuel}(t) = 1.86 - 0.367 \cdot t - 0.0669 \cdot t^2 \quad (Kg) \quad (1.15)$$

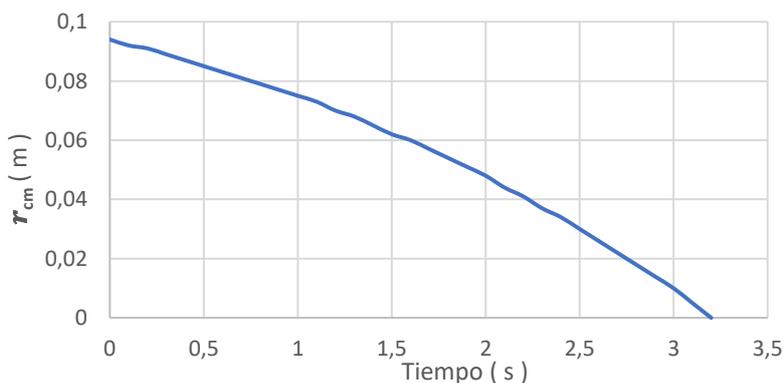
A partir de la ecuación de la masa se calculan sus derivadas respecto al tiempo para obtener \dot{m} y \ddot{m} .

1.6.5.3. Posición del Centro de Gravedad

Como hay variación de masa, también variará la posición del centro de gravedad (CDG) por el tiempo. Este parámetro es necesario ya que se toman momentos respecto del centro de dry mass (masa en vacío), y si la posición del centro de gravedad no coincide con la de el centro de dry mass, existirán momentos a tener en cuenta.

Así pues, para calcularlo se ha hecho uso del software OpenRocket, que introduciendo el esquema del cohete ofrece la posición del centro de gravedad del mismo. Para poder obtener una ecuación de la variación de este, se han tomado los datos del centro de gravedad para 32 valores de masa del motor, la cual se ha ido modificando manualmente en el programa. Los 32 valores corresponden a cada instante de tiempo múltiplo de 0.1 s, y se ha calculado el valor de la masa en ese instante según las ecuaciones 1.13 y 1.15.

Una vez obtenidos los datos, se ha interpolado la curva que forman los datos de la posición del CDG en metros frente al tiempo en segundos (Gráfica 1) mediante mínimos cuadrados y se ha obtenido la siguiente función con un error de mínimos cuadrados de $5.6 \cdot 10^{-6}$.



Gráfica 1: Curva de la posición del CDG frente al tiempo

$$r_{cm} = \begin{cases} 0,093 - 0,013 \cdot t - 0,00493 \cdot t^2 & 0 < t < 3,2 \\ 0 & 3,2 < t \end{cases} \quad (1.16)$$

Cabe destacar que, al considerar simetría en dos planos, el cambio del centro de gravedad solo afecta al eje vertical, mientras que en los ejes \widehat{b}_1 y \widehat{b}_2 se considera que no hay variación. Así pues, la fórmula 1.16 se correspondería a la coordenada z del vector de movimiento del centro de gravedad, siendo las demás coordenadas igual a 0.

1.6.5.4. Tensor de Inercia

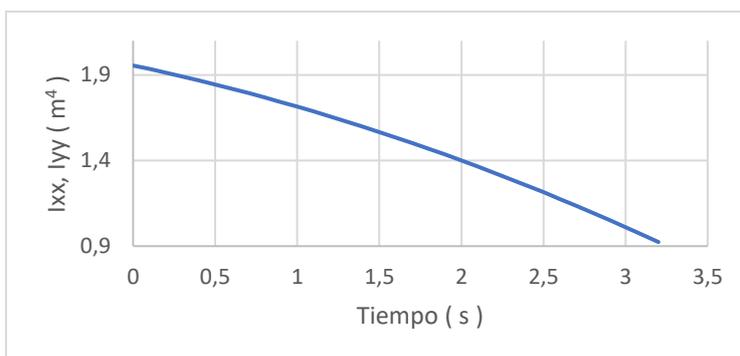
Este parámetro genera bastante dificultad para ser calculado analíticamente por la gran cantidad de elementos que componen el cohete, es por eso por lo que se suele utilizar un programa de diseño 3D para representar el cohete y sacar el tensor con las herramientas del software. Sin embargo, a la fecha de realización de este trabajo no se dispone de un fichero CAD del ensamblaje del cohete, ni se poseen datos fiables del peso que van a tener cada una de las partes del cohete.

Para poder calcular este dato se ha hecho una maqueta del cohete en Autodesk Fusion 360 sin incluir todos los elementos internos, simplemente separando el cohete en dos partes, la parte que corresponde al motor y el resto. De esta manera, se han ajustado los pesos de cada una de estas partes. Se ha fijado como masa de la parte superior la masa en vacío del cohete completo menos la masa del motor en vacío. Así, para definir la masa de la parte correspondiente al motor, se ha calculado para cada instante de tiempo la masa de fuel y se ha sumado a la masa del motor en vacío.



Figura 18: Maqueta de ASTRA con separación por pesos

De este modo, mediante las herramientas del software, se obtiene la matriz de inercia para cada valor de masa. Dichos datos se almacenan y, siguiendo un proceso similar al del apartado anterior, se interpolan todos los valores de I_{xx} , los de I_{yy} y los de I_{zz} , para obtener la función de cada uno de ellos respecto del tiempo. Como ya se ha comentado, se considera que $I_{xy} = I_{xz} = I_{yz} = 0$.



Gráfica 2: Gráfica de I_{xx} e I_{yy} en función del tiempo

Nos damos cuenta de que los valores de I_{xx} e I_{yy} son iguales. Esto se debe a que, en esas direcciones, el cohete es completamente simétrico, por lo que no existe ningún factor que altere uno y no el otro. Esto es así por la consideración tomada de que el cohete es un cilindro perfecto, debido a la maqueta utilizada. Esto, con un diseño CAD correcto (fichero que se obtendrá por parte del equipo en el futuro próximo), podría no ocurrir.

Así pues, interpolando las 3 curvas de datos y obteniendo unos errores de mínimos cuadrados del orden de 10^{-8} se obtiene el siguiente tensor de inercia respecto al tiempo.

$$\mathbf{I} = \begin{cases} \begin{pmatrix} 1,9 - 0,2 \cdot t - 0,04 \cdot t^2 & 0 & 0 \\ 0 & 1,9 - 0,2 \cdot t - 0,04 \cdot t^2 & 0 \\ 0 & 0 & 0,004 - 3,7 \cdot 10^{-4} \cdot t - 8,3 \cdot 10^{-5} \cdot t^2 \end{pmatrix} & t < 3,2 \\ \begin{pmatrix} 0,924 & 0 & 0 \\ 0 & 0,924 & 0 \\ 0 & 0 & 0,0018 \end{pmatrix} & t \geq 3,2 \end{cases} \quad (1.17)$$

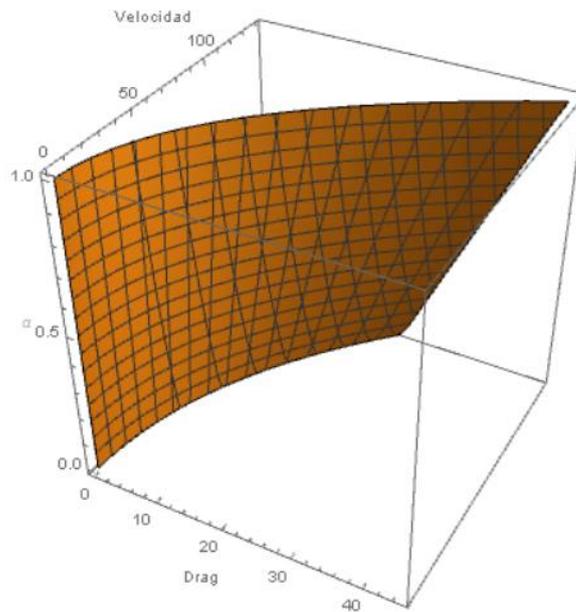
Derivando esta ecuación en función del tiempo obtenemos el parámetro \mathbf{I}' .

1.6.5.5. Expresión del Drag

El Drag es una fuerza que es aplicada siempre en la dirección tangente a la trayectoria y en sentido contrario. Es por esto que, en el body-frame, el drag solo presentará un valor distinto de 0 en el eje \widehat{b}_3 .

Como se ha mencionado, el estudio aerodinámico de los aerofrenos y el cohete ha sido realizado por otro miembro del equipo, quien proporcionó una tabla de datos del Drag total del cohete dependiendo de 2 variables, la velocidad del cohete y el porcentaje de salida de los aerofrenos en tanto por 1.

La tabla contenía 500 valores de Drag utilizando 100 valores de velocidad y 5 valores de porcentaje de salida de aerofrenos: 0, 0.25, 0.5, 0.75 y 1.



Gráfica 3: Gráfica 3D del Drag en función de la velocidad y alfa

Como es de esperar, a mayor velocidad y mayor porcentaje de aerofrenos, mayor drag es generado. Además, se puede ver que cuando V vale 0, el Drag es igual a 0 independientemente del porcentaje de salida de los aerofrenos, esto se podía deducir de antemano ya que si el cohete no está en movimiento no existe ninguna fuerza aerodinámica actuando sobre él.

Así pues, usando el procedimiento de mínimos cuadrados, se han interpolado los datos de la Gráfica 3 para obtener la siguiente expresión del Drag.

$$D = 5.698 \cdot 10^{-4} \cdot V \cdot \alpha + 1.13 \cdot 10^{-3} \cdot V^2 + 5.55 \cdot 10^{-3} \cdot V + 1.07 \cdot 10^{-3} \cdot V^2 \cdot \alpha \quad (1.18)$$

Donde α es el porcentaje de salida de los aerofrenos y V es la velocidad del cohete.

Cabe destacar que este valor de Drag se aplica únicamente en la dirección tangente a la trayectoria, es decir, en el eje \widehat{b}_3 .

1.7. Implementación en Matlab Simulink

1.7.1. Definición de los parámetros

El primer paso para poder introducir el modelo en Simulink es definir el valor de los parámetros que son constantes durante todo el vuelo. Estos parámetros son la aceleración de la gravedad y la posición de la tobera respecto al centro de dry mass. Estos parámetros se añaden previamente al modelo en Simulink añadiéndose al workspace para que Simulink reconozca el valor de las variables al introducirlas como bloques “Constant”. El código para introducir las constantes es muy sencillo:

```
%% Parámetros simulink
rnoz=[0 0 0.986];
g=9.807;
```

Figura 19: Código de implementación de las constantes

El siguiente paso para introducir el modelo en Simulink es escribir las ecuaciones de los parámetros. La manera más simple de introducir funciones en Simulink es mediante el bloque “Matlab Function”, que te permite escribir en código las características de la función y especificar el número de entradas y salidas.

Modelo Propulsivo.

Se adjunta el esquema y código de obtención del parámetro T para cada instante de tiempo t.

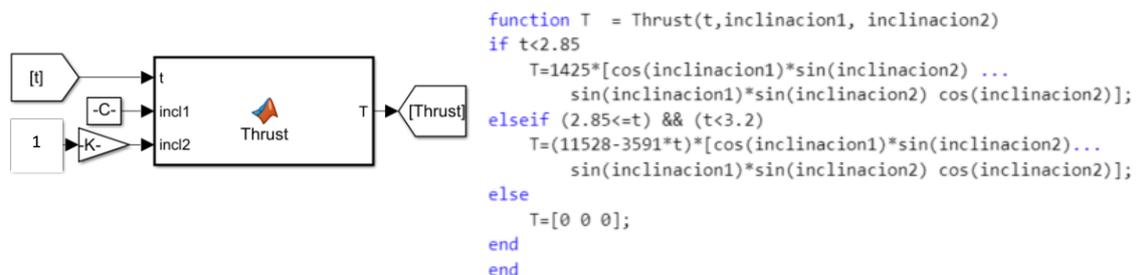
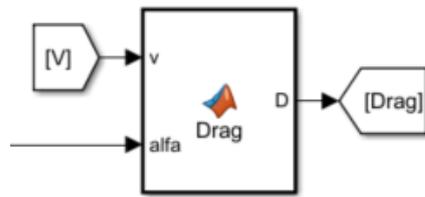


Figura 20: Modelo propulsivo en Simulink

Donde -C- es el valor de 45° en radianes que corresponde al valor de β y -K- es un conversor de grados a radianes, que cambia el valor de α introducido en el bloque anterior en grados.

Ecuación del Drag

De manera similar se introduce la ecuación del Drag, que depende de la velocidad y del porcentaje de salida de los aerofrenos, llamado “alfa”. La velocidad introducida es la velocidad calculada por el modelo, que se introduce en el modelo del drag para obtener la velocidad en el instante siguiente. Por otro lado, el valor de *alfa* se introduce mediante la salida del controlador que se va a explicar en los siguientes apartados. Por el momento basta comentar que la entrada alfa del Drag es la variable a controlar del modelo. El código y disposición de este parámetro queda de la siguiente manera.



```
function D = Drag(v,alfa)
Drag=0.00056979*v(3)*alfa+0.001127223*v(3)^2+...
0.005548636*v(3)+0.001074272*v(3)^2*alfa;

D=[0 0 Drag];

end
```

Figura 21: Modelo del Drag en Simulink

Posición del centro de gravedad

En la definición de la posición del centro de gravedad se va a introducir también la implementación de sus derivadas, ya que como se ha visto son necesarias en las ecuaciones del modelo. También se definen en forma de vector en el body-frame, por lo que teniendo en cuenta las simetrías las coordenadas x e y de los vectores sería 0.



```
function r3 = r3(t)
if t<3.2
    r3=[0 0 -(0.093179393-...
0.013054469*t-0.004926939*t^2)];
else
    r3=[0 0 0];
end
end
```



```
function dr3 = dr3(t)
if t<3.2
    dr3=[0 0 -(-0.013054469-...
2*0.004926939*t)];
else
    dr3=[0 0 0];
end
end
```



```
function d2r3 = d2r3(t)
if t<3.2
    d2r3=[0 0 -(-2*0.004926939)];
else
    d2r3=[0 0 0];
end
end
```

Figura 22: Implementación de la posición del CDG y derivadas

Ecuaciones de la masa

Igual que en apartado anterior, también se incluyen en este apartado la implementación de las derivadas de la masa, parámetros necesarios en las ecuaciones a desarrollar. En este caso, la masa y sus derivadas se define como escalar y únicamente dependen del tiempo.

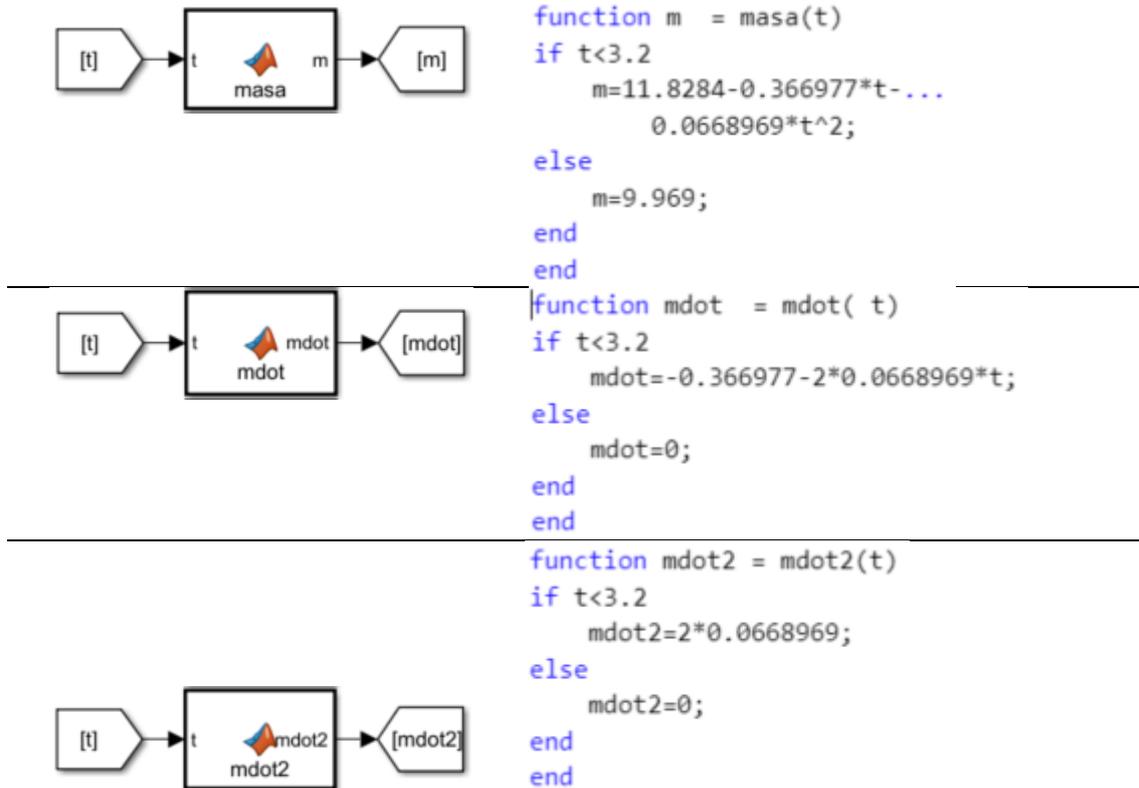


Figura 23: Implementación de la masa y derivadas

Tensor de Inercia

El tensor de inercia y su derivada dependen únicamente del tiempo, y la salida de la función es el tensor de inercia entero, es decir, una matriz donde los elementos de la diagonal corresponden a I_{xx} , I_{yy} e I_{zz} , y los demás términos son 0 como se ha comentado.

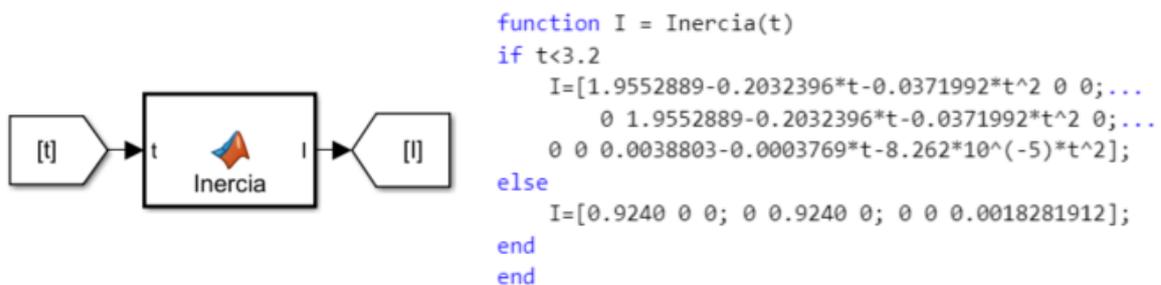


Figura 24: Implementación del Tensor de Inercia

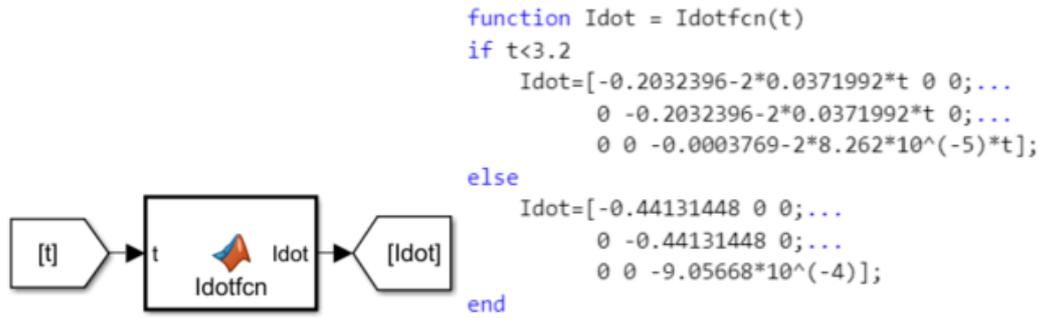


Figura 25: Implementación de la derivada del Tensor de Inercia

Matriz de Rotación

Por último, se introduce la función de rotación, que depende de los ángulos de Euler. Estos ángulos nos los proporciona el modelo en cada instante de tiempo al ser una de las variables de estado. La salida se proporciona en forma de matriz 3x3.

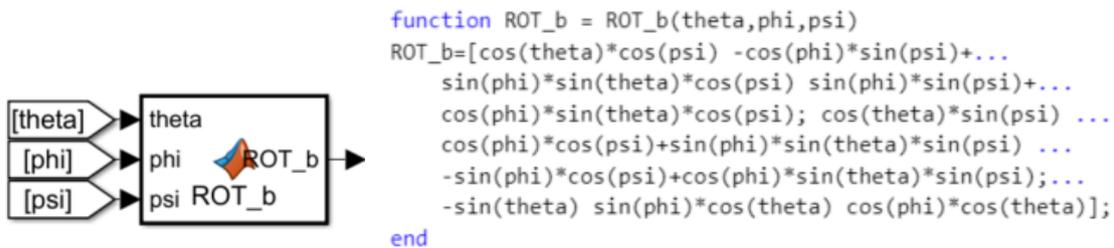


Figura 26: Implementación de la matriz de rotación

1.7.2. Funciones adicionales

Para el correcto funcionamiento de la simulación se necesitan una serie de funciones de desarrollo. Estas funciones ya existen dentro de la librería de Simulink, sin embargo, no presentan un comportamiento correcto por la naturaleza del modelo. Son funciones que son incompatibles por motivos de tamaño y longitud de matrices y vectores, cosa que provoca problemas en el momento de simular en algunas de sus aplicaciones en el modelo. Dichas funciones son el producto escalar de una matriz por un vector, y el producto vectorial.

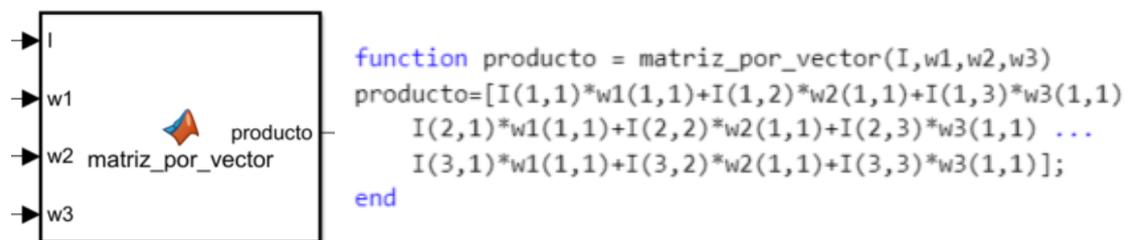


Figura 27: Implementación del producto escalar

Donde el parámetro I corresponde a la matriz que se desea multiplicar y w1, w2 y w3 representan cada uno de los elementos de un vector de tres dimensiones.

Como se puede ver, para lograr el funcionamiento correcto se ha tenido que especificar más rigurosamente el elemento concreto que se está utilizando en cada momento en las ecuaciones, aparte de tener que introducir elemento a elemento cada componente del vector que se desea multiplicar. Esto debe ser así ya que en su comportamiento normal se interpretaban los tamaños de los vectores de manera errónea.

Por otro lado, se introduce la definición del producto vectorial.

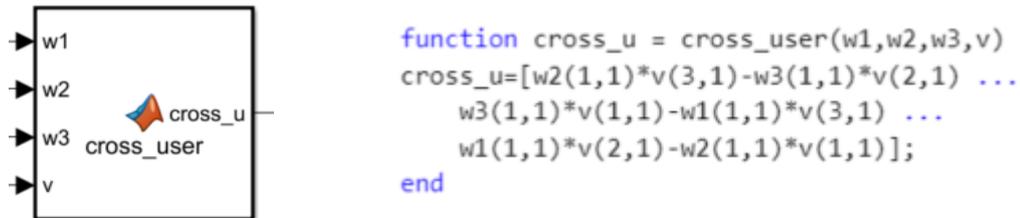


Figura 28: Implementación del producto vectorial

Donde v corresponde a uno de los vectores a multiplicar y, al igual que en la función anterior, $w1$, $w2$ y $w3$ son los elementos del otro vector.

Del mismo modo que en el producto escalar, en esta implementación se debe introducir el vector elemento a elemento, y especificar concretamente qué elemento se está usando en cada término de la ecuación.

1.7.3. Ecuación de Fuerzas

Una vez definidos todos los parámetros que se van a utilizar y las funciones necesarias, se pasa a introducir las ecuaciones del modelo para poder realizar la simulación.

El primer paso es reestructurar la ecuación de Fuerzas para poder obtener todas las variables de estado que pretendemos obtener de esta ecuación. Para obtenerlas, como la posición \mathbf{r} no aparece en la ecuación de manera explícita, se tiene que integrar la velocidad \mathbf{v} para obtenerla. Además, para simplificar el montaje de las ecuaciones en Simulink, esta variable \mathbf{v} también se va a obtener de la integración de la aceleración $\dot{\mathbf{v}}$. De este modo, para obtener las dos variables de estado \mathbf{r} y \mathbf{v} se va a reestructurar la ecuación de Fuerzas para aislar la aceleración lineal $\dot{\mathbf{v}}$. La ecuación 1.6 entonces quedaría de la siguiente manera.

$$\dot{\mathbf{v}} = \frac{1}{m} (\mathbf{T} - 2\dot{m}\mathbf{r}_{CM}' + 2\boldsymbol{\omega} \times (\mathbf{r}_{noz} - \mathbf{r}_{CM}) + \ddot{m}(\mathbf{r}_{noz} - \mathbf{r}_{CM}) - \mathbf{D} - R_A^B(mg\hat{\mathbf{a}}_3) + m \cdot (-\dot{\boldsymbol{\omega}} \times \mathbf{r}_{CM} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{CM}) - \mathbf{r}_{CM}'' - 2\boldsymbol{\omega} \times \mathbf{r}_{CM}')) \quad (1.6.1)$$

De esta manera se simplifica la implementación en Simulink, organizando la ecuación en dos sumatorios, uno multiplicado por la variable m y otro que incluye todos los términos de la ecuación. Finalmente, para obtener las variables de estado, se divide el resultante por m y se integra el resultante, una vez para obtener \mathbf{v} y dos veces para obtener \mathbf{r} .

Sumatorio 1

El primer sumatorio (segunda línea de la ecuación 1.6.1) se introduce haciendo uso de los bloques que proporciona Simulink. En concreto se van a utilizar los bloques “Cross”, que realiza el producto vectorial de los dos vectores que se proporcionan como entrada; “Gain”, que multiplica la entrada por una constante que se debe especificar; “Sum”, que permite introducir el número de términos a sumar especificando el signo; y “Product”, que multiplica las entradas entre sí, pudiendo especificar el número de entradas.

Así pues, se adjunta, siguiendo el mismo orden, el esquema en Simulink del término de la ecuación de la segunda línea de 1.6.1.

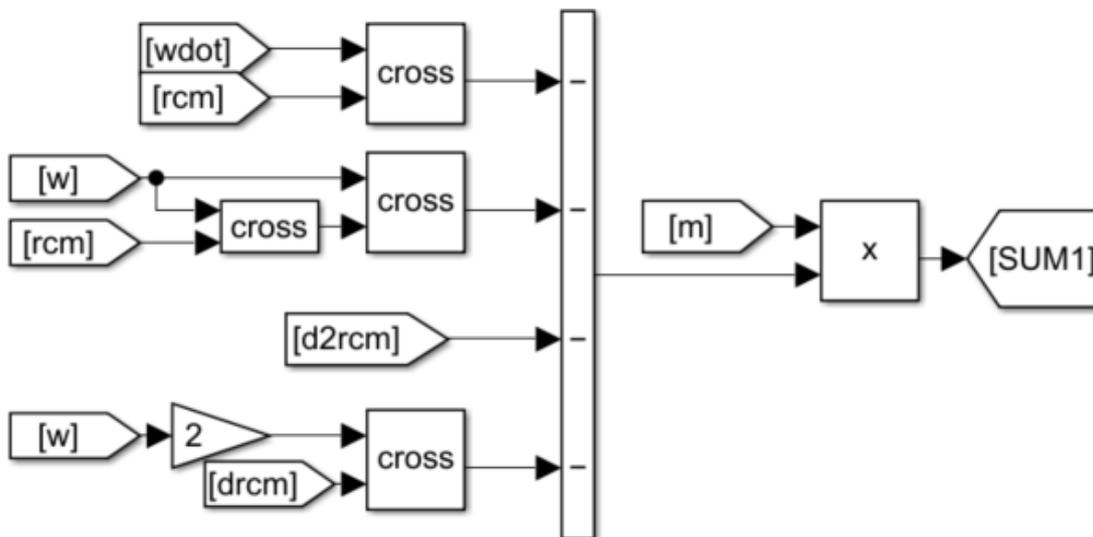


Figura 29: Implementación en Simulink del Sumatorio 1

Donde las etiquetas “wdot”, “w”, “rcm”, “drcm”, “d2rcm” y “m” hacen referencia respectivamente a las variables $\dot{\mathbf{w}}$ y \mathbf{w} y los parámetros \mathbf{r}_{CM} , \mathbf{r}'_{CM} , \mathbf{r}''_{CM} y m .

Sumatorio 2

El segundo sumatorio incluye todos los términos de la ecuación, por lo que la salida del mismo será la aceleración lineal. Una vez introducidos todos los términos en este sumatorio, se integrará la salida para obtener la velocidad lineal y se volverá a integrar para obtener la posición.

Para poder implementar esta parte se han usado los bloques de Simulink siguientes: “Cross”, “Sum” y “Product”, explicados anteriormente; “Invert 3x3 Matrix”, que realiza la inversa de una matriz 3x3, utilizada para invertir la matriz de transformación de sistemas de referencia; “Constant”, donde se introduce el valor de una constante; y “Divide”, que funciona como un bloque “Sum” pero con multiplicaciones y divisiones.

Además, para integrar la señal de salida del sumatorio se utiliza el bloque “Integrator”, que la segunda vez que se utiliza se le añade una opción que permite limitar la salida para que solo permita valores positivos en la coordenada Z.

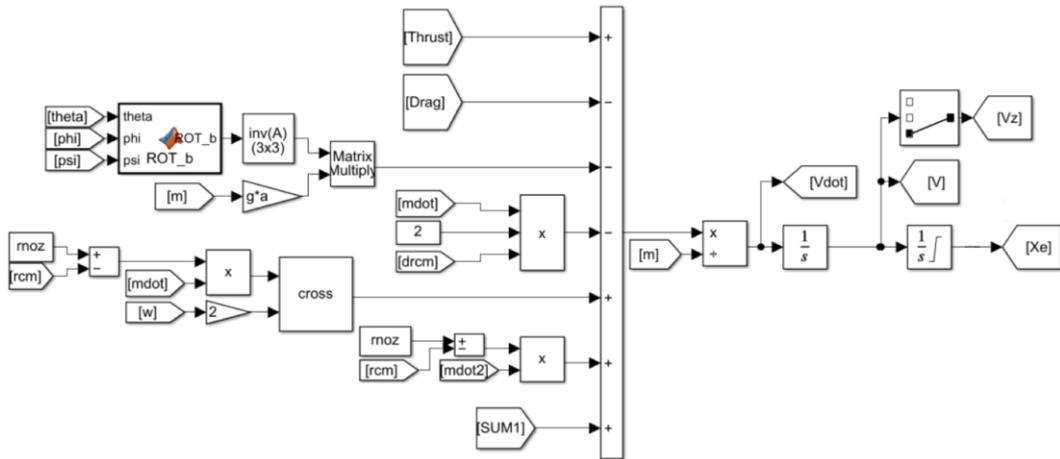


Figura 30: Implementación en Simulink del Sumatorio 2 e integración

Se tienen que explicar además el uso de algunas etiquetas y secciones. La etiqueta “Vz” se obtiene con el bloque “Selector”, un bloque que permite aislar uno de los componentes de una señal vector, en este caso, se utiliza para obtener el valor de la velocidad vertical.

Además, las etiquetas “theta”, “phi” y “psi” hacen referencia a los ángulos de Euler. Estas etiquetas se obtienen de la ecuación de momentos que se explicará en el siguiente apartado, de donde se saca la variable de estado \mathbf{q} y mediante el siguiente esquema utilizando el bloque “Selector” se aíslan cada uno de los ángulos.

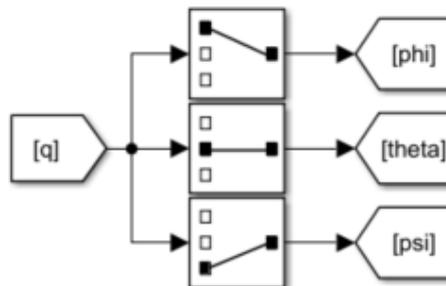


Figura 31: Esquema de obtención de los ángulos de Euler

1.7.4. Ecuación de Momentos

De forma análoga se tiene que introducir la ecuación de momentos. Como la variable \mathbf{q} no aparece de forma explícita en la ecuación, esta se debe obtener mediante la integración de la variable \mathbf{w} , que, a su vez, para facilitar el montaje, esta se va a obtener de la integración de la variable $\dot{\boldsymbol{\omega}}$.

Así pues, la mejor manera de obtener las variables de estado es aislando la variable de la aceleración angular e integrándola dos veces. La ecuación de momentos quedaría de la siguiente manera:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(-\boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega}) - \mathbf{I}' \cdot \boldsymbol{\omega} - m\mathbf{r}_{CM} \times \dot{\mathbf{v}} - \mathbf{r}_{CM} \times mg\hat{\mathbf{a}}_3 + \mathbf{M}_{est}) \quad (1.11.1)$$

Así pues, se organiza la ecuación agrupando todos los términos a un lado del igual y multiplicándolos por la inversa del tensor de inercia por la izquierda. Cabe destacar que en esta ecuación es donde se usan las funciones adicionales comentadas en el apartado 1.7.2. Estas ecuaciones se necesitan porque con los bloques habituales que ofrece Simulink aparecen unos errores en las dimensiones de los vectores a utilizar. Para solventar este error, se han creado estas funciones donde se tienen que especificar cada uno de los componentes del vector que genera esos problemas que es el vector w . Para aislar esos componentes se utiliza un esquema similar a la Ilustración 22, pero que además dichos elementos se vuelven a juntar en un solo vector mediante el bloque “Mux”, que permite unir diversas variables en un solo vector. Esa variable reunida es la que se usa en la ecuación de Fuerzas.

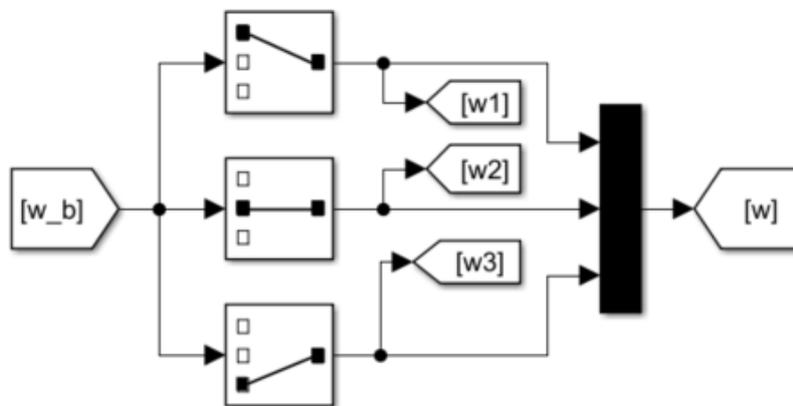


Figura 32: Esquema de separación y reunión de los elementos de w

Donde “ w_b ” es la variable sacada de la ecuación de momentos que generaba problemas de compilación.

Momentos Inerciales

El término de momentos inerciales se introduce mediante los bloques “Cross” y “Product” ya explicados.

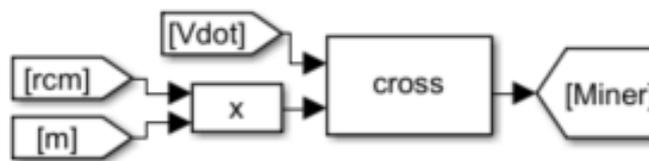


Figura 33: Implementación de los momentos inerciales

Momento angular

El término de momentos angulares está dividido, ya que uno de los términos es el propio término de la aceleración angular que se ha aislado. Para introducirlo, se han introducido los términos no afectados por la variable ω en un sumatorio, utilizando las funciones alternativas y bloques comunes ya explicados.

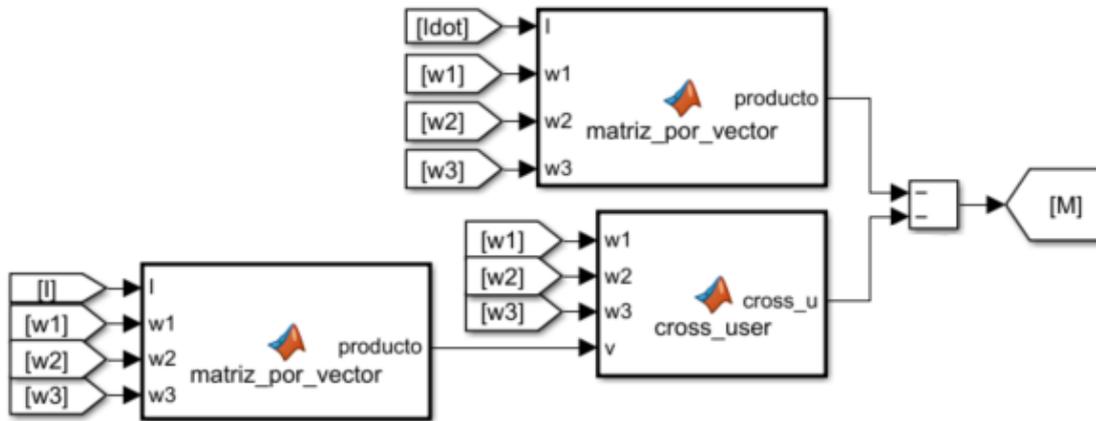


Figura 34: Implementación de los momentos angulares

Como se puede ver, para el uso de las funciones alternativas se especifican los elementos de la velocidad angular por separado, que se han obtenido de la Ilustración 23.

Momentos Externos

Para el término de momentos externos hay dos términos muy distintos. Por un lado, está el término inducido por la gravedad y por otro lado el término de los momentos del estabilizador. El término gravitatorio es trivial y se introduce con bloques básicos ya explicados. Sin embargo, el término de los estabilizadores es un momento regulado por dos controladores PID. El desarrollo y ajuste de dichos controladores se realizará en el apartado de control de este documento.

De este modo, se adjunta la implementación de este término.

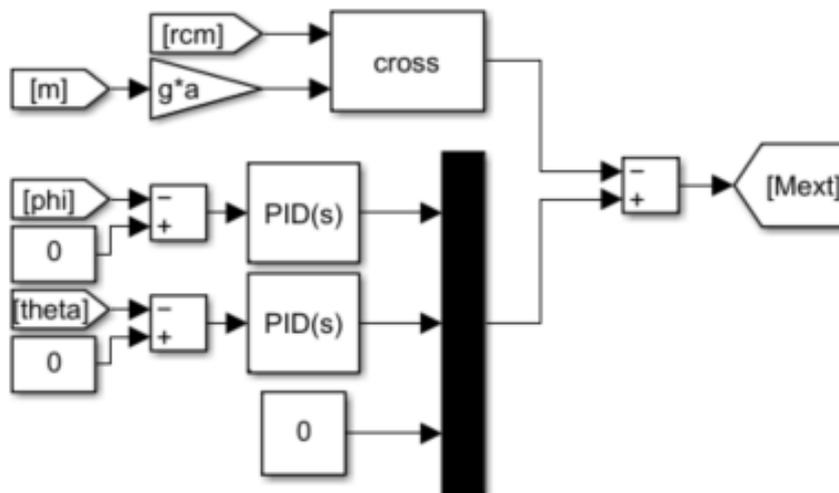


Figura 35: Implementación de los momentos Externos

Como se puede ver, los controladores PID comparan los ángulos de Roll y Pitch del modelo con el valor 0 para lograr la estabilización mediante controladores activos con dichos ángulos nulos. No se pretende controlar el Yaw ya que no existe ningún componente en las ecuaciones que le afecte, por lo que, en este modelo, esta variable será nula durante todo el vuelo.

Sumatorio principal

De esta manera, se introducen todos los términos en un sumatorio principal. La saliente de ese sumatorio se multiplica por la izquierda por la inversa del tensor de inercia como introduce la ecuación. De ese modo, surge la aceleración angular y se integra para obtener la velocidad angular (término que luego se separará en sus componentes según el esquema de la Ilustración 23) y se vuelve a integrar para obtener la variable de estado q .

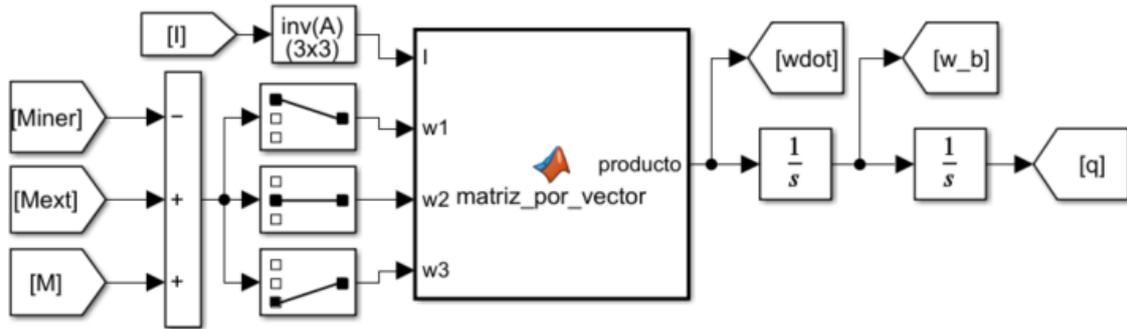


Figura 36: Implementación de la ecuación de Momentos completa

Como se puede ver, también se ha tenido que utilizar la función alternativa de producto escalar para no obtener errores de compilación.

1.7.5. Esquema del control de altitud

En este apartado se explicará simplemente el esquema que sigue la implementación en Simulink de lo necesario para realizar el control de altitud, pero no se explicaran en detalle las características del control, su funcionamiento o los parámetros necesarios para controlar el apogeo.

Para realizar el control se necesita una señal de referencia. La forma de sacar dicha señal se explica en el siguiente apartado. Esa señal de referencia pasa por un bloque “Selector” para obtener la coordenada Z de la misma y se compara con la señal del modelo. Esa información se introduce en el controlador PID, cuya salida está limitada a valores entre 0 y 1 ya que en ese rango se encuentran los valores posibles que pueden tomar los aerofrenos en tanto por uno de salida. Además, se introduce un bloque “Switch”. Este bloque permite seleccionar qué acción se lleva a cabo dependiendo de una condición que se introduce en la entrada central. En este caso, se utiliza para poder determinar en qué casos se efectúa el control y en qué casos no.

En nuestro caso, solamente nos interesa accionar los aerofrenos en la fase de ascenso, puesto que el apogeo solo es controlable en esa etapa. A partir de llegar a la altitud máxima, nos interesa que los aerofrenos se recojan para no interferir negativamente en labores posteriores como el despliegue del paracaídas.

De ese modo, se introduce en el bloque “Switch” la condición de que sólo bajo valores de velocidad positivos se active el control, en caso contrario $\alpha=0$.

El esquema queda de la siguiente manera:

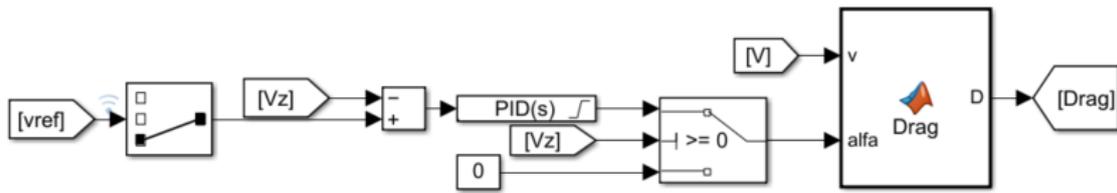


Figura 37: Esquema del sistema de control de apogeo

1.7.6. Señal de referencia

La señal de referencia para el controlador de altitud se introduce mediante un subsistema dentro del esquema de Simulink. Un subsistema, bloque “Subsystem”, es un bloque donde se puede introducir un archivo esquema Simulink completo del que se sacan unas variables determinadas. Se puede interpretar como una función Matlab ya utilizada, pero en lugar de código se introduce la expresión deseada mediante bloques Simulink.

Dentro del subsistema destinado a la obtención de la señal de referencia, que en este caso es la velocidad como se explicará en el apartado de control, se introduce exactamente el mismo modelo que en el sistema principal, solo que con un cambio. La diferencia entre el subsistema y el sistema principal es el valor del Drag. En el subsistema, no se pretende controlar el porcentaje de salida de los aerofrenos, sino que se pretende proporcionar un perfil de velocidad que consiga hacer que la trayectoria llegue a una altura determinada. De este modo, la única variante del subsistema es que el Drag ha sido modificado para no depender de los aerofrenos, y se ha ajustado la ecuación para que se genere un Drag que haga alcanzar una altitud máxima determinada.

Ese valor de Drag debe ser uno que permita llegar a una altitud intermedia entre la altitud máxima a la que se llega con los aerofrenos desplegados al 100% desde el primer instante y la altitud de apogeo a la que se llega sin despliegue de aerofrenos. De este modo, desplegando los aerofrenos un porcentaje intermedio, se llegará a una altura intermedia, y esa es la que tenemos que introducir en el subsistema. El esquema quedaría del siguiente modo, donde la entrada es el tiempo y la salida es la velocidad de referencia.



Figura 38: Esquema del Subsistema

Dando doble clic sobre este bloque se abre un lienzo Simulink en blanco donde se introduce el mismo modelo que en el lienzo principal con las modificaciones comentadas.

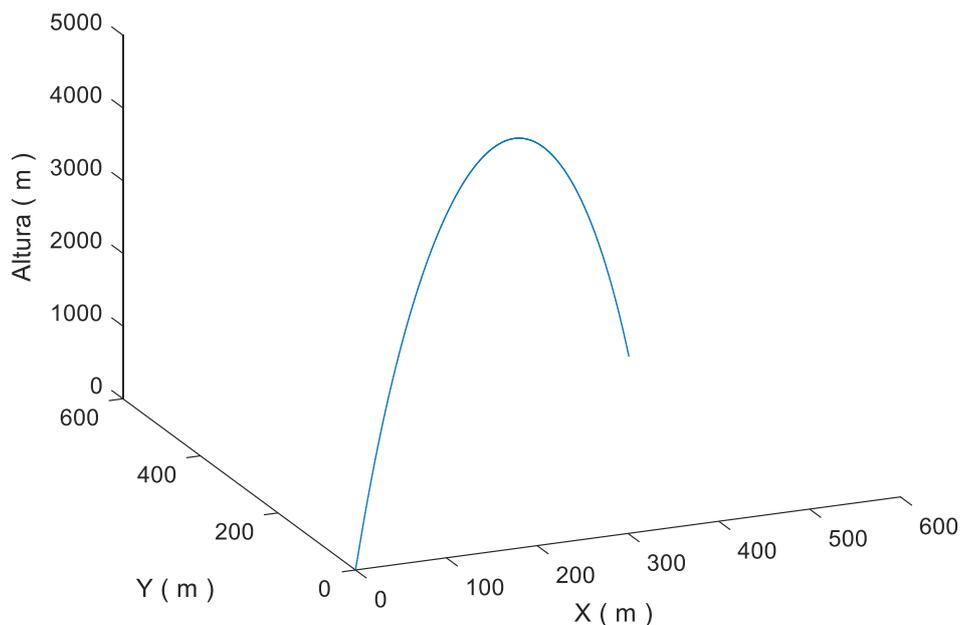
1.8. Simulación

Una vez introducidos todos los parámetros en el archivo de Simulink, ya es posible realizar una simulación del vuelo. Cabe destacar que no se va a disponer de una animación donde se pueda visualizar el cohete y su movimiento, sino que lo que se va a visualizar son las gráficas en función del tiempo de cada parámetro que interese.

Los datos que más nos interesan son los del movimiento. Se va a querer visualizar la trayectoria y en específico la altura para poder ver con precisión el apogeo del vuelo.

Para esta primera simulación, al no tener los parámetros del PID para poder realizar el control sobre los aerofrenos, se van a establecer dos valores fijos de aerofrenos, 0 y 1, para visualizar el apogeo en los dos casos extremos, para primeramente comprobar que el apogeo deseado de 4000 metros se encuentra entre esos valores, lo que querría decir que se puede realizar el control.

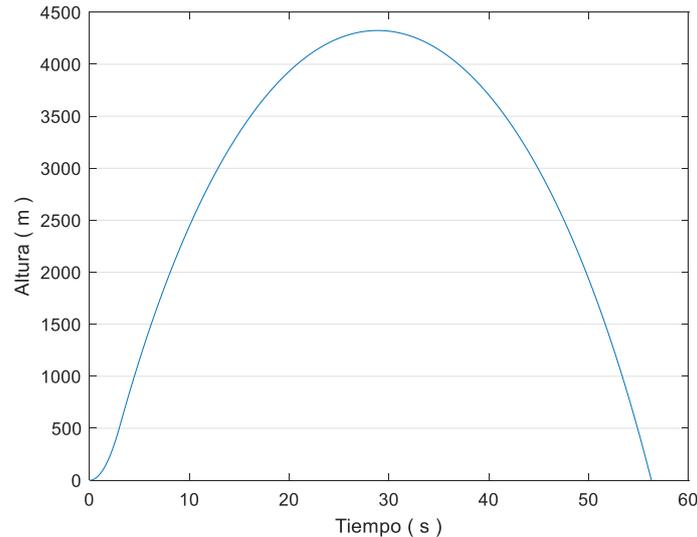
Así pues, se adjunta la gráfica de la trayectoria para un vuelo sin aerofrenos. Con esta gráfica se pretende visualizar sobre todo el desplazamiento lateral del cohete.



Gráfica 4: Trayectoria sin aerofrenos

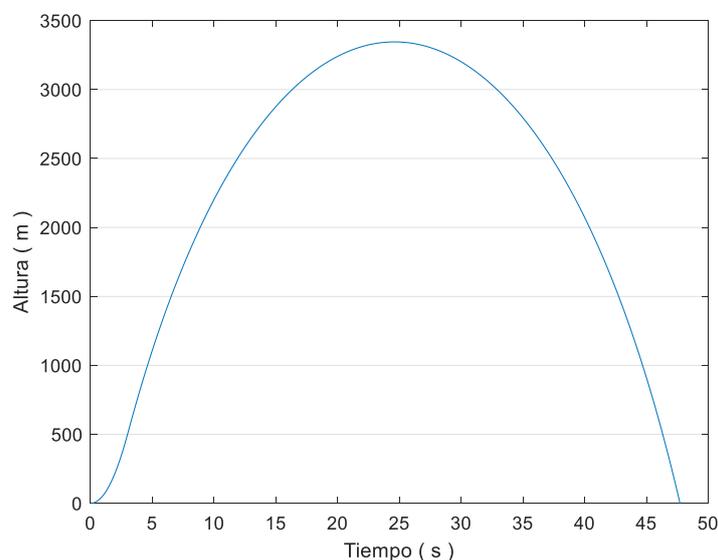
Como se puede ver, en cada uno de los ejes del plano horizontal se llega a unos valores de entorno a los 500 metros. Este desplazamiento lateral se debe a la inclinación introducida como imperfección en la orientación del motor. Esa inclinación introduce una cantidad de empuje en dichos ejes que generan ese pequeño desplazamiento.

Además, se va a obtener la gráfica separada de la altura para visualizar mejor el apogeo. En esta gráfica se va a visualizar el mayor valor de apogeo que logra el cohete, puesto que no se introducen los aerofrenos. Es por esto por lo que se espera que supere los 4000 metros para luego poder frenarlo con el control hasta esa cifra.



Gráfica 5: Altura de la trayectoria sin aerofrenos

Como se puede ver, se sobrepasan los 4000 metros por lo que se logra el límite superior de apogeo que da libertad para frenarlo hasta esa altitud. Sin embargo, hay que comprobar que se puede frenar lo suficiente como para reducir el apogeo hasta los 4000 metros. Para ello, se volverá a simular con los aerofrenos desplegados al máximo desde el primer instante para ver si el apogeo queda por debajo de esa altura, lo que querría decir que existe capacidad de frenar el cohete suficientemente, e incluso en exceso para disponer de libertad en la acción de control.



Gráfica 6: Altura de la trayectoria con aerofrenos al máximo

Así pues, al no llegar a los 4000 metros con los aerofrenos desplegados al máximo, podemos asegurar que se puede conseguir un control que frene el cohete la cantidad justa como para llegara exactamente a la altitud deseada. Habiendo comprobado que se puede lograr el objetivo con el modelo introducido, se puede empezar con el diseño de los sistemas de control automático.

1.9. Control Automático

1.9.1. Principios del Control Automático

Un sistema de control automático es un conjunto de componentes conectados entre sí que permiten regular una o varias características de su propio comportamiento de manera autónoma, es decir, sin intervención de agentes externos al sistema. Además, dicho sistema puede contener sensores que detecten el estado del sistema para corregirlo comparando el comportamiento deseado con el comportamiento obtenido.

La mejor forma de representar un sistema de control es mediante la representación en diagramas de bloques. Un diagrama de bloques se puede obtener como un circuito con un flujo de información representado por flechas que circulan por una disposición de bloques que representan las diferentes características del sistema. Un sistema sin control automático seguiría el siguiente esquema:

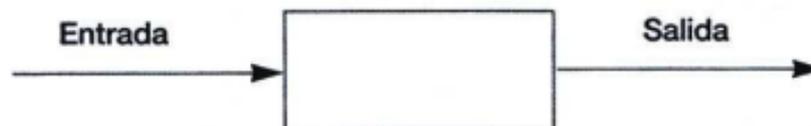


Figura 39: Diagrama de bloques de un sistema simple (ISHUELIN, 2022)

Como se puede ver, la representación nos indica en qué dirección se mueve la entrada, donde se manipula esa entrada, y en qué dirección se proporciona la salida. Estos diagramas de bloques se pueden complicar hasta el punto de representar ecuaciones complejas con ellos, ya que contienen sus propias reglas y operaciones para hacer con ellos. Sin embargo, con propósitos académicos se tiende a englobar un sistema complejo que no tiene interés didáctico en un solo bloque que expresa la naturaleza de todo el sistema.

Existen dos tipos principales de sistemas de control, de lazo abierto y de lazo cerrado. Los sistemas de lazo abierto son los sistemas de control en los que la acción de control no depende de la salida del controlador, es decir, que la señal de salida no influye en la señal de entrada. Estos sistemas dependen de la calibración del controlador, ya que no poseen un sistema que detecte el error y corrija la acción de control, sino que el mismo controlador debe realizar el control de la señal de entrada para obtener la señal esperada sin posibilidad de reajuste del controlador. El diagrama de bloques de un sistema de control de lazo abierto es:



Figura 40: Diagrama de bloques de un sistema de lazo abierto (ISHUELIN, 2022)

Por otro lado, existen los sistemas de lazo cerrado. En estos sistemas la acción de control depende de la salida del controlador. En este proyecto, se va a utilizar este tipo de sistemas ya que en nuestro caso la acción de los aerofrenos afecta a la velocidad del cohete, que es el parámetro de entrada al controlador. Este modo de introducir como entrada la señal de salida modificada por el sistema del modelo se le llama realimentación. Es por esto que a los sistemas de lazo cerrado también se definen como sistemas en los que existe una realimentación de la señal de salida. El diagrama de bloques de un sistema realimentado sigue el siguiente esquema:

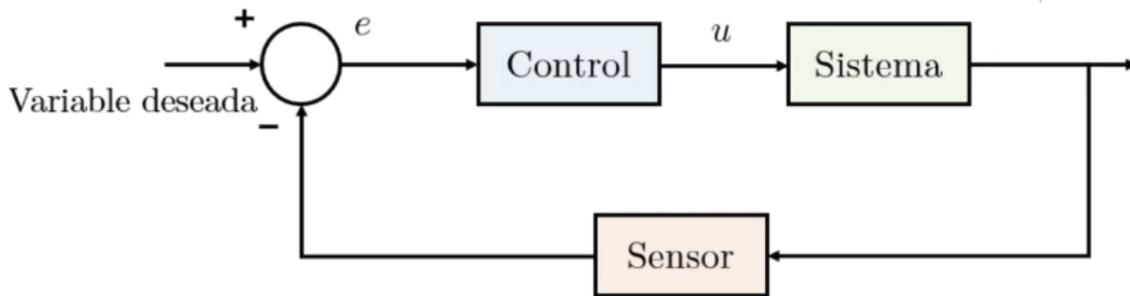


Figura 41: Diagrama de bloques de un sistema de lazo cerrado (Christian Trejo | Sistemas Dinámicos y Control, 2021)

Como se puede ver, la variable deseada, también llamada referencia, se introduce en un sumador donde se compara con la salida del sistema, que nos proporciona el valor obtenido de esa variable. A la salida del sumador se le denomina error, ya que nos dice cuánto se desvía la señal obtenida de la señal deseada. Ese error pasa por el sistema de control, que ofrece un valor en la variable o variables de control que pretende lograr que, tras pasar por el sistema, la salida de ese sistema tenga error 0 respecto a la señal deseada.

Este tipo de sistemas sí son capaces de asegurar que se logra la señal deseada, mientras que en los sistemas de lazo abierto no siempre se obtiene, sino que existen perturbaciones. Así pues, el objetivo es ajustar el controlador para que el error entre la señal obtenida y la señal deseada sea nulo.

1.9.1.1. Definición de FdT y polos y ceros del sistema

Una FdT (Función de Transferencia) es una expresión matemática que por medio de un cociente relaciona la respuesta de un sistema con una señal de entrada. Dicha relación se realiza en el dominio de Laplace o dominio de frecuencia. Así pues, la función de transferencia establece la relación entre las transformadas de Laplace de las señales de entrada y salida del sistema.

El denominador de una FdT se denomina función característica, y las raíces de la función característica igualada a 0 se conocen como polos del sistema. Mientras que las raíces de la ecuación del numerador se denominan polos del sistema.

Los polos y ceros de un sistema determinan su LDR (Lugar de las Raíces). El LDR es un gráfico en dos dimensiones correspondientes al eje real y el eje imaginario en el que se representa el lugar geométrico de los polos de la función de transferencia. (ISHUELIN, 2022)

1.9.1.2. Controladores PID

Un controlador PID es un mecanismo de control de lazo cerrado. Consta de tres elementos, el Proporcional, el Integrador y el Derivativo.

El valor del proporcional se dice que depende del presente, es decir, intenta controlar el error en el instante de tiempo actual. El valor del integrador se basa en el pasado ya que recoge los errores que han ocurrido previamente para realizar el control. El valor derivativo depende del futuro, lo que significa que intenta prever el error futuro y actúa en consecuencia. (WIKIPEDIA, 2022. *Controladores PID*)

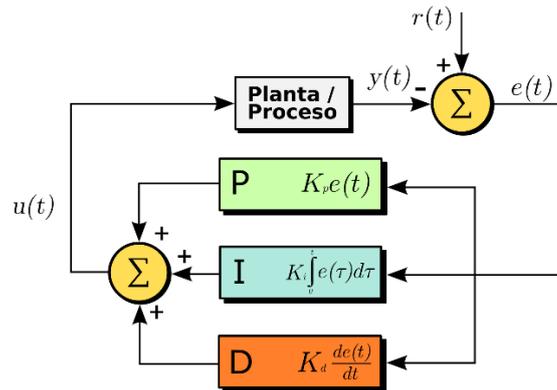


Figura 42: Diagrama de un sistema con control PID (WIKIPEDIA, 2022. *Controladores PID*)

Como se puede ver, el Diagrama 9 es igual al Diagrama 8 donde se detalla el bloque de controlador desglosándolo en los parámetros del PID, donde se especifican las ecuaciones que lo rigen en el dominio temporal. Estas ecuaciones se pueden pasar al dominio de Laplace que es donde se va a realizar el control. Quedarían de la siguiente manera:

$$P = kc \qquad I = \frac{kc}{Ti \cdot s} \qquad D = s \cdot kc \cdot Td \quad (1.19)$$

Así pues, ajustando los valores de cada uno de estos parámetros, se puede modificar el comportamiento del controlador hasta lograr los resultados deseados. Para ello, existen una serie de procedimientos y reglas que ayudan al ajuste del controlador.

La regla principal es que el controlador PID debe ser lo más sencillo posible, atendiendo al siguiente orden de preferencia: P→PI→PD→PID. Además, para guiar en qué tipo de sistema es más correcto, se sigue el esquema del Diagrama 10.

Además, se pueden demostrar las siguientes expresiones que rigen el comportamiento de cada tipo de controlador.

$$PI = k \frac{(s+a)}{s} \quad (1.20)$$

$$PD = k(s + b) \quad (1.21)$$

$$PID = k \frac{(s+a)(s+b)}{s} \quad (1.22)$$

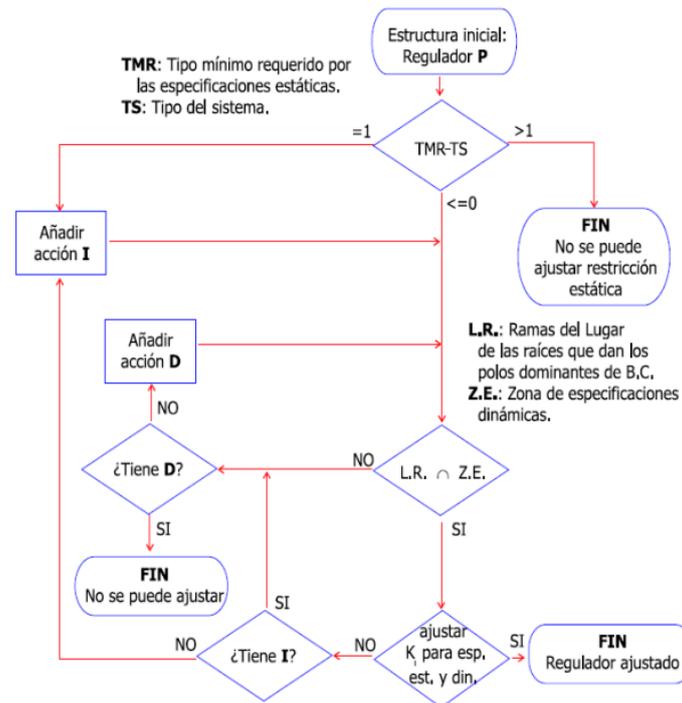


Figura 43: Esquema del procedimiento para el ajuste de un PID (UPV, 2020)

Donde el tipo de sistema se refiere al número de integradores (s en el denominador) que posee la FdT del proceso y el Tipo Mínimo Requerido depende de la cantidad y tipo de error que se quiera lograr en el control, en el caso de este proyecto, al desear error nulo, el TMR es igual a 1.

1.9.2. Obtención de la FdT

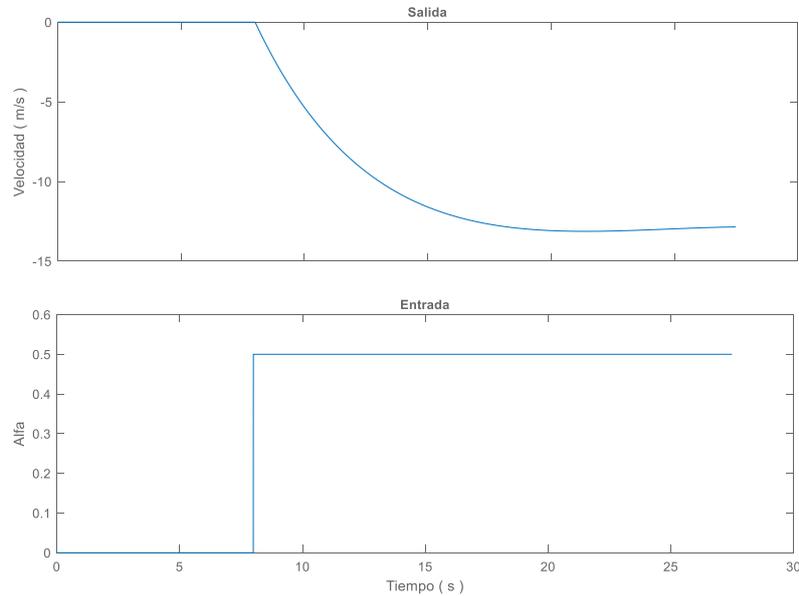
Para conocer la Función de Transferencia del modelo debemos conocer la respuesta del sistema ante entradas de tipo escalón. Para ello el procedimiento es someter el sistema a dos entradas diferentes y compararlas entre si, y de esa respuesta obtener la FdT.

En este caso, la entrada del sistema es el valor de los aerofrenos, por lo que se va a visualizar la salida del sistema ante dos funciones de tipo escalón en los aerofrenos. En nuestro caso, la salida del sistema la vamos a considerar la velocidad del cohete en el eje vertical, ya que, como se ha visto, la señal de referencia es un perfil de velocidad ideal que provoca el apogeo deseado.

Para facilitar el cálculo, se van a usar una serie de herramientas que ofrece Matlab Simulink. En primer lugar, para introducir la función escalón se va a introducir el bloque “Step”, y posteriormente se usará el paquete System Identification Toolbox para estimar la FdT en función de los datos obtenidos.

Las dos señales de entrada que se van a usar son una función escalón de valor 0.5 en el instante $t=8$ s, y un valor constante de 0. Con estas entradas, se obtienen los dos perfiles de velocidad y se restan entre si para compararlos. Dichos datos se introducen en la aplicación systemIdentification para estimar la FdT.

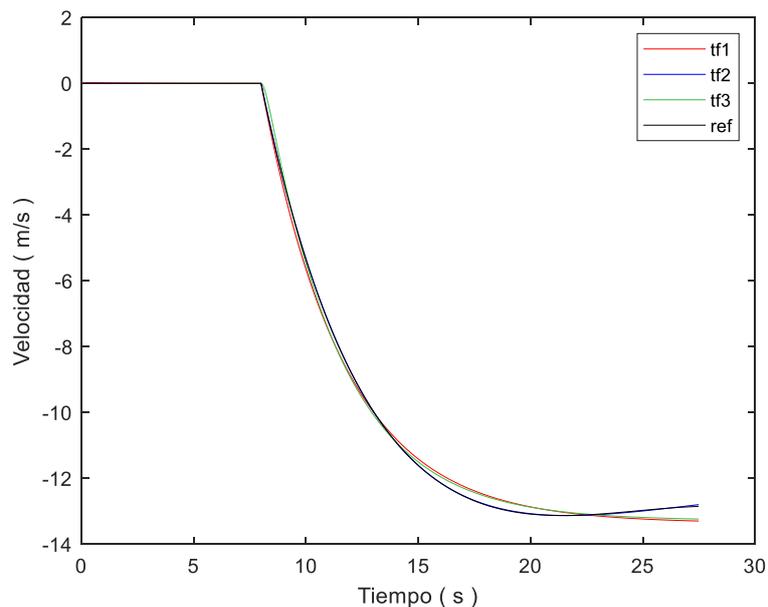
Así pues, las señales de entrada y salida que nos van a servir para estimar la FdT se muestran en la siguiente imagen, donde se utilizan simplemente los datos hasta el momento del apogeo ya que ese es el proceso que nos interesa controlar.



Gráfica 7: Señales de salida y entrada respectivamente

Como se puede ver, la señal de salida se mantiene en valor 0 hasta el instante en el que se introduce la función escalón, ya que hasta ese instante hay el mismo valor de aerofrenos por lo que la resta de ambos perfiles será 0.

Así pues, se realizan varias estimaciones de la FdT mediante las herramientas disponibles para lograr el mejor ajuste posible. En este caso, se han realizado tres estimaciones: tf1, con un polo y ningún cero; tf2, con dos polos y un cero; y tf3 con dos polos y ningún cero.



Gráfica 8: Ajuste de las diferentes estimaciones

Como se puede ver, la FdT que más se ajusta a la referencia es la $tf2$, concretamente el programa nos especifica un ajuste del 99.79, frente al 96.99 de $tf3$ y el 96.51 de $tf1$. Esta Función de Transferencia es la siguiente:

$$tf2 = G(s) = \frac{-6.52 \cdot s - 0.6436}{s^2 + 0.3024 \cdot s + 0.02716} \quad (1.23)$$

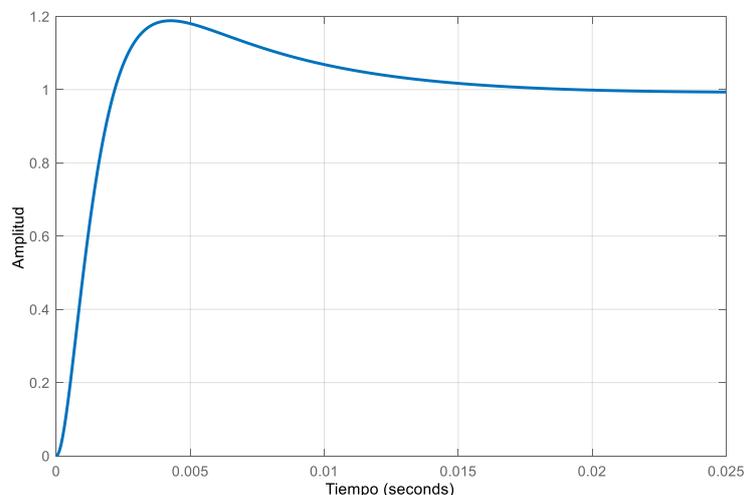
1.9.3. Ajuste del PID de los estabilizadores

Como se ha explicado, el momento M_{est} corresponde a un momento que generan los estabilizadores del cohete que actúa en sentido contrario a la dirección de w , es decir, que mantiene el cohete con ángulos de Euler cercanos a cero.

Estos estabilizadores actúan de manera pasiva, aprovechando su forma y los efectos aerodinámicos que generan para generar esos momentos compensatorios cuando aparece una desestabilización. Sin embargo, el modelaje de estos estabilizadores es una tarea muy costosa y que se escapa de los objetivos de este proyecto, por lo que no se va a pretender modelar los estabilizadores de manera pasiva, sino que se han añadido dos controladores PID que regulan la estabilización de manera activa, añadiendo el momento correcto para estabilizar el cohete y mantener los ángulos de Euler a 0.

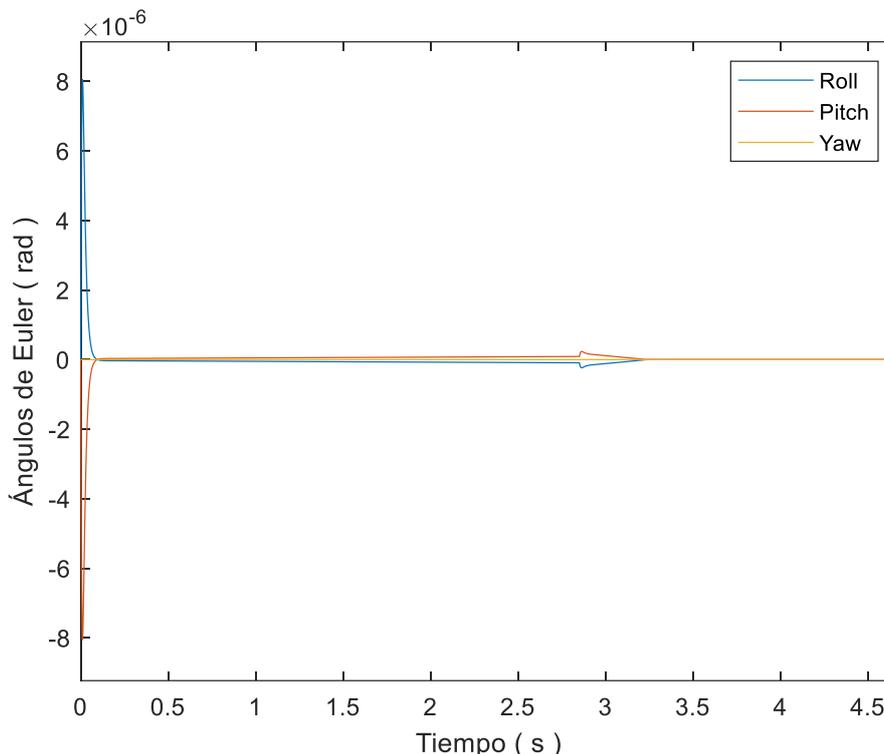
Para ajustar los PID, se ha utilizado la herramienta pidTuner del paquete de Matlab “Simulink Control Design”. Esta herramienta establece automáticamente los parámetros de un PID introducido en un modelo Simulink. Utilizando métodos numéricos, crea un modelo lineal de el sistema no lineal introducido, y con esa aproximación ajusta el PID para obtener los resultados deseados. En la interfaz de usuario se nos permite modificar el tiempo de establecimiento y la sobreoscilación de la respuesta.

Así pues, modificando estos parámetros se ha introducido en el PID los valores que ofrecen una repuesta más rápida y con menor sobreoscilación que permiten la estabilidad del sistema. Una vez establecidos, se obtiene la siguiente Step Response (el significado de esta gráfica está explicado en el apartado 1.9.4 en la Gráfica 12).



Gráfica 9: Step Response del PID de los estabilizadores

Así pues, es de interés visualizar el comportamiento de los ángulos de Euler para comprobar el funcionamiento del momento M_{est} .



Gráfica 10: Ángulos de Euler

Como se puede ver, alrededor del segundo 3 se estabilizan completamente los ángulos de Euler a valor 0. Sin embargo, las pequeñas oscilaciones que hay son del orden de 10^{-6} . Aunque son valores muy cercanos a 0, dichos valores son durante el tiempo que el motor está encendido, por lo que van a afectar en la trayectoria del cohete ya que durante esos instantes se introducen unas fuerzas en los ejes del plano vertical adicionales a las que se introducen por la inclinación introducida en el motor que no se van a contrarrestar durante todo el vuelo.

Así pues, el comportamiento de los ángulos de Euler confirma la completa estabilidad del cohete a la par que introduce pequeños valores de empuje adicionales en los ejes horizontales que van a resultar en una contribución en el desplazamiento lateral.

En los experimentos reales, dichos valores de inclinación serían más elevados y no se corregirían tan rápidamente. Se obtendrían oscilaciones mayores que se traducirían en un desplazamiento lateral bastante más cuantioso que el que se obtiene en este modelo.

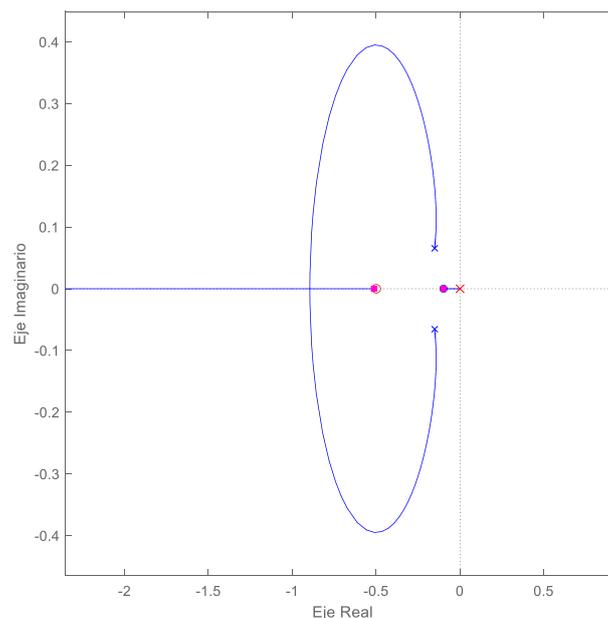
En el lanzamiento de “ASPERA”, que se realizó hasta altitudes mucho menores, se obtuvo un desplazamiento lateral de aproximadamente el 50% del valor de apogeo que alcanzó, por lo que se puede prever que en el lanzamiento del futuro cohete “ASTRA” se obtendrán proporciones similares entre el desplazamiento lateral y vertical si se realizan en condiciones similares de viento y situación atmosférica.

1.9.4. Ajuste del PID de control de apogeo

Primeramente, cabe destacar que el ajuste de este controlador no se puede realizar con la herramienta pidTuner como el control de estabilidad ya que el modelo no lineal es demasiado complejo como para linealizarlo con los métodos que utiliza esta herramienta. Así pues, se van a usar los métodos habituales mediante la función de transferencia calculada en el apartado 1.9.2 para ajustar el controlador.

Para ello, se va a utilizar el Diagrama 10 como guía para el procedimiento. Partiendo de un controlador proporcional, el primer paso es saber qué rama tomar del diagrama. Para ello, se debe hacer la resta entre el tipo mínimo requerido para obtener el error deseado, que al querer error nulo el tipo mínimo para lograrlo es tipo 1, y el tipo de sistema, que al no tener ningún integrador en $G(s)$ el sistema es de tipo 0. Así pues, el diagrama especifica que se debe añadir acción integral. Como se ha visto, un controlador PI se corresponde con la función $k \frac{(s+a)}{s}$, en la que se tiene que especificar el valor de la posición del cero, a , y la ganancia k .

La manera recomendada de determinación de ceros en los controladores es mediante cancelación de polos. Sin embargo, como el proceso contiene dos polos imaginarios, no se puede cancelar uno sin cancelar el otro ya que no se pueden introducir ceros asimétricos con respecto al eje real en el LDR, es decir, si se introduce un cero complejo en un controlador tiene que ser junto a su conjugado. Sin embargo, si se introducen dos ceros complejos, el controlador pasaría a ser de la forma $k \frac{(s+a)(s+b)}{s}$, lo cual hemos visto que se corresponde con un controlador PID. Así pues, atendiendo al criterio de construcción de controladores simples, antes de introducir dos ceros por cancelación obteniendo un PID, se va a introducir un único cero (junto al integrador) para obtener un controlador PI. En concreto, se ha establecido el cero en $s = -0.5$, se adjunta el LDR.

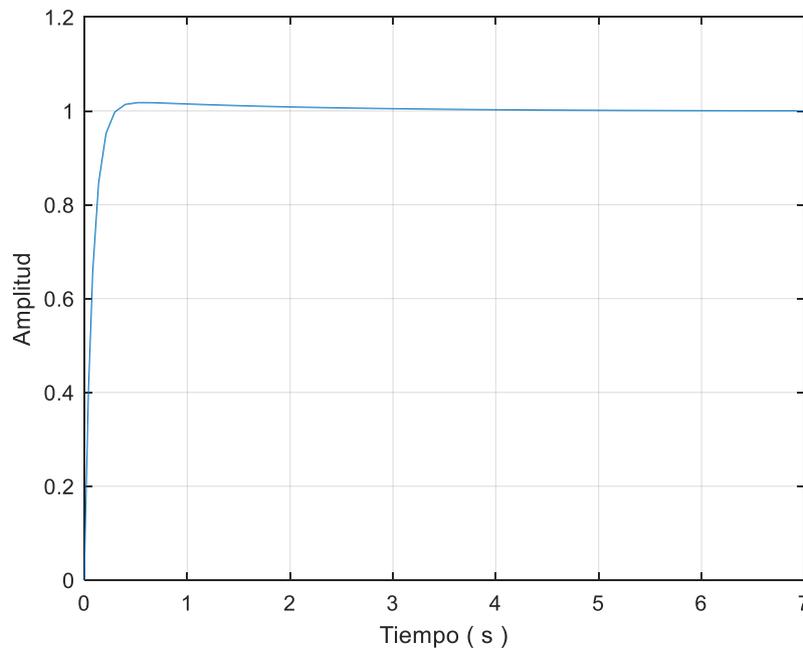


Gráfica 11: LDR del proceso controlado

Donde se puede ver el polo en $s = 0$ correspondiente al integrador, el polo en $s = -0.5$, los dos polos imaginarios y el cero de $G(s)$. Así pues, para ajustar el controlador simplemente se tiene que ajustar la ganancia k . Para ello, como se pretende una respuesta rápida del sistema, con poco error y con rapidez en la estabilización en error 0, se debe añadir mucha ganancia. Así pues, se añade un valor de ganancia de -2. De esta manera, la función del controlador C queda de la siguiente manera:

$$C = -2 \cdot \frac{(s + 0.5)}{s} \quad (1.24)$$

Además, es de interés visualizar la respuesta del sistema con este controlador ante una entrada escalón de amplitud 1. En esta gráfica se espera ver poca o nula oscilación, ya que tener mucha oscilación se traduciría en un movimiento demasiado variado de los aerofrenos cosa que podría resultar en su rotura; y un tiempo de establecimiento bajo, para obtener con presteza el error nulo y poder reducir el máximo el movimiento de los aerofrenos.

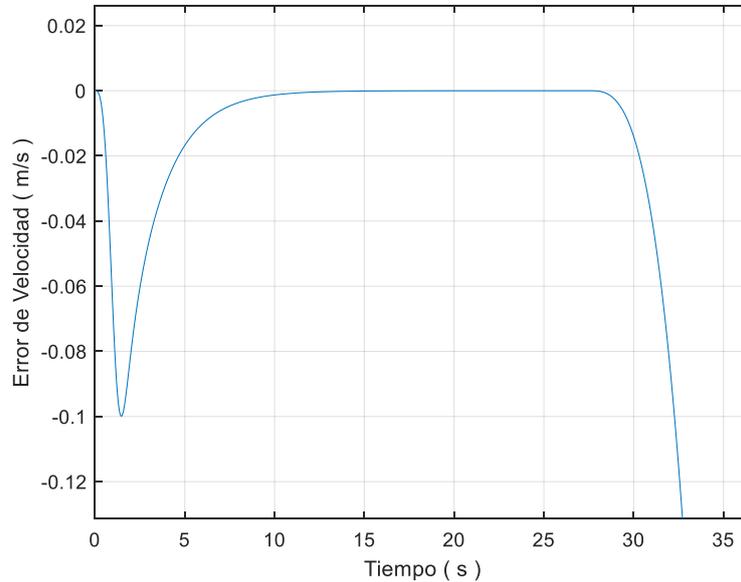


Gráfica 12: Step Response del modelo controlado

Como se puede ver, se logra la oscilación muy baja y un tiempo de establecimiento suficientemente pequeño, por lo que se considera un controlador válido a priori, debiéndose comprobar su funcionamiento mediante la simulación y obtención de resultados.

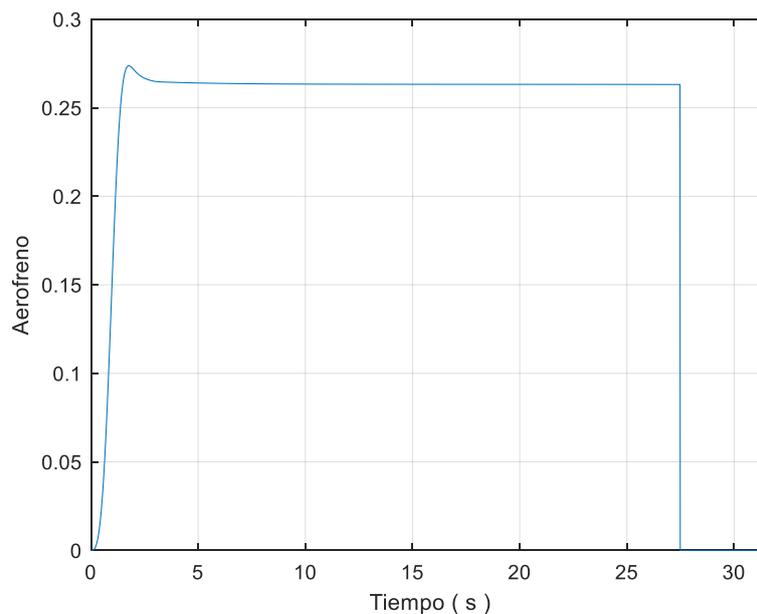
Para ello, se deben introducir los valores de P e I en el controlador de Simulink. Desarrollando la ecuación anterior se obtienen los parámetros P e I que se deben introducir en el controlador: $C = -2 - \frac{1}{s}$. Así pues, atendiendo a la ecuación 1.19 se observa que $P = -2$ e $I = -1$. Introduciendo estos valores en el controlador PID del modelo en Simulink, se comprueba su funcionamiento simulando.

Para considerar exitoso el ajuste del controlador, se espera la estabilización rápida del error entre la velocidad del modelo y la velocidad de referencia en valor 0. Además, también se debe producir la misma sin oscilación o con oscilación significativamente pequeña por temas de resistencia estructural de los aerofrenos. Finalmente, por motivos obvios se debe lograr error 0 entre la referencia y el modelo llegando a la altura máxima de 4000 metros. Para comprobarlo, se obtiene las gráficas del error.



Gráfica 13: Error de velocidad

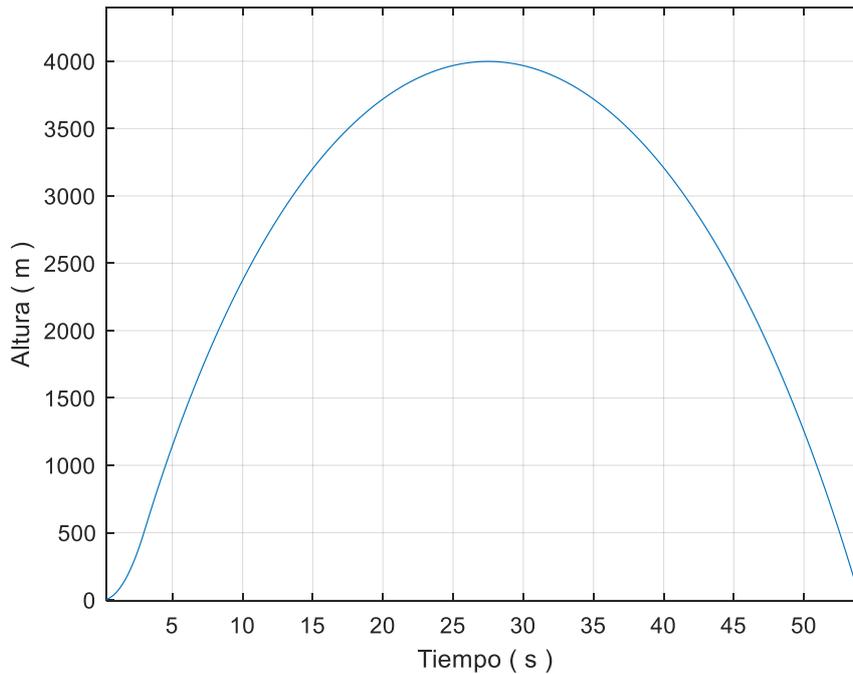
Como se puede ver, el error de velocidad se estabiliza en 0 alrededor del segundo 15, haciendo previamente un pico de valores suficientemente bajos, por lo que el control es efectivo, además de conseguir sobreoscilación nula. Para comprobar que la acción del aerofreno no es excesivamente rápida o brusca pudiendo provocar fallos físicos en la estructura, se visualiza su gráfica.



Gráfica 14: Acción del aerofreno

Como se puede ver, tarda alrededor de 3 segundos en alcanzar el valor del 27% aproximadamente, de lo que se puede deducir que no es un movimiento excesivamente rápido, y por lo tanto asumible por los aerofrenos sin fallos estructurales. Además, hay algo de oscilación, pero significativamente pequeña, por lo que no se tendrán problemas a la hora de lograrla en la realidad.

Finalmente, para comprobar el funcionamiento completo, se visualiza la gráfica de la altura de la trayectoria para ver si efectivamente se llega a la altura de apogeo deseada.



Gráfica 15: Altura de la trayectoria

Así pues, al haber llegado a la altura deseada de 4000 metros, con error nulo y un movimiento asumible estructuralmente por los aerofrenos, se concluye el ajuste del PID de control de altitud y por lo tanto se da por logrado el objetivo principal de este documento.

1.10. Conclusiones

Volviendo al objetivo principal de este proyecto, que era diseñar un sistema de control automático que regulara con precisión la altitud máxima del vuelo de un cohete sonda diseñado por Faraday Rocketry UPV. Dicho objetivo ha sido cumplido con error nulo y con un rango de movimientos asequible por el futuro sistema físico que se va a implementar en el cohete “ASTRA”.

Cabe destacar algunas consideraciones que se han tenido para aclarar en qué condiciones se ha realizado este control. En primer lugar, no se han tenido en cuenta algunos componentes aerodinámicos como la sustentación aerodinámica. Este efecto modificaría la trayectoria tanto horizontal como verticalmente. La falta de estos efectos genera un perfil perfecto, sin momentos angulares, por lo que la trayectoria debería surgir de forma puramente vertical, es decir, sin desplazamiento lateral. Para obtener una trayectoria más cercana a la realidad, se ha introducido deliberadamente un error en la colocación de los motores como medio de obtener algo de desplazamiento lateral, es decir, introducir imperfecciones deliberadas en el modelo para acercarse al resultado final.

Además, tampoco se han modelado los estabilizadores del cohete. Los estabilizadores son una parte muy importante del cohete ya que son el único elemento que se dispone de asegurar la estabilidad del mismo, de no ser por ellos el cohete con alta probabilidad se desestabilizaría bruscamente dando vueltas sobre su eje. Para contrarrestar la falta de estos en el modelo, se ha introducido un controlador activo para asegurar la estabilidad, sin embargo, este control es mucho más rápido que lo que probablemente sucedería en la realidad, además de que no contribuye en la sustentación aerodinámica ni en otros efectos aerodinámicos como sí lo hacen los estabilizadores reales.

Por último, la suposición de atmósfera en calma es bastante desviada de la realidad, ya que lo normal es tener viento en el momento del lanzamiento y es un factor que influye significativamente en la trayectoria del cohete. Sin embargo, al no poder predecir el comportamiento del viento el día del lanzamiento, se debe introducir una dirección y potencia de viento elegida por el usuario, cosa que puede estar muy desviada de la realidad, y por eso se ha decidido no incluirlo en el modelo

Con todas estas consideraciones se ha conseguido diseñar un modelo cercano a la realidad y se ha logrado el ajuste del controlador necesario para obtener el objetivo deseado, por lo que se considera exitoso el diseño del sistema de control de apogeo del cohete “ASTRA” bajo ciertas suposiciones.

1.11. Referencias

WIKIPEDIA, 2021. *Cohete sonda* [en línea]. [Consulta: 25 de Julio 2022]. Disponible en: https://es.wikipedia.org/wiki/Cohete_sonda

WIKIPEDIA, 2022. *Cohete* [en línea]. [Consulta: 2 de Julio 2022]. Disponible en: <https://es.wikipedia.org/wiki/Cohete>

BLOGSPOT, 2013. *Fuegos artificiales* [en línea]. [Consulta: 3 de Julio 2022]. Disponible en: <http://historiainventos.blogspot.com/2013/09/fuegos-artificiales.html>

LIVEJOURNAL, 2008. *The othmer library of chemical history* [en línea]. [Consulta: 3 de Julio 2022]. Disponible en: <https://bibliodyssey2lj.livejournal.com/25945.html>

PUBLICDOMAINVECTORS.ORG, 2017. *Diagrama del cohete* [en línea]. [Consulta: 4 de Julio 2022]. Disponible en: <https://publicdomainvectors.org/es/vectoriales-gratuitas/Diagrama-del-cohete/57166.html>

NTRGUADALAJARA, 2020. *El nuevo y exitoso enfoque para recuperar cohetes espaciales* [en línea]. [Consulta: 4 de Julio 2022]. Disponible en: https://www.ntrguadalajara.com/post.php?id_nota=158643

WIKIWAND, 2022. *Mercury Redstone 1ª* [en línea]. [Consulta: 4 de Julio 2022]. Disponible en: https://www.wikiwand.com/es/Mercury_Redstone_1A

ELCONFIDENCIAL, 2016. *Así funciona la revolucionaria tecnología de Elon Musk* [en línea]. [Consulta: 4 de Julio 2022]. Disponible en: https://www.elconfidencial.com/tecnologia/2016-05-06/space-falcon-9-aterrizaje-tecnologia_1190434/

WIKIPEDIA, 2022. *Motor cohete* [en línea]. [Consulta: 8 de Julio 2022]. Disponible en: https://es.wikipedia.org/wiki/Motor_cohete

VIDEOFROMSPACE, 2021. *SpaceX Starship SN10 soars, lands for first time!*. En: *Youtube.com* [video en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://www.youtube.com/watch?v=gLNQ6Mq5kbg>

ISHUELIN, 2022. *Sistemas Automáticos de Control* [en línea]. [Consulta: 19 de Julio 2022]. Disponible en: <http://ieshuelin.com/huelinwp/download/Tecnologia/Tecnologia%20industrial/3-SISTEMAS-AUTOMaTICOS-DE-CONTROL-ampliacion.pdf>

Christian Trejo | Sistemas Dinámicos y Control, 2021. *Introducción a los Sistemas de Control en Lazo Cerrado | Ejemplo de Esquema de Control de Velocidad* En: *Youtube.com* [video en línea]. [Consulta: 20 de Julio 2022]. Disponible en: <https://www.youtube.com/watch?v=T5F89iH6ndU>

WIKIPEDIA, 2022. *Controlador PID* [en línea]. [Consulta: 20 de Julio 2022]. Disponible en: https://es.wikipedia.org/wiki/Controlador_PID



UPV, 2020. *Control Automático* [en línea]. [Consulta: 21 de Julio 2022].
Disponible en: <https://poliformat.upv.es/>

THRUSTCURVE, Cesaroni 4263L1350-P. En: ThrustCurve.org [en línea].
Disponible en <https://www.thrustcurve.org/motors/Cesaroni/4263L1350-P/> [Consulta:
29 de Marzo de 2022]

CEOTTO, Giovanni, et al., 2021. *RocketPy: Six Degree-of-Freedom Rocket Trajectory Simulator*. [en línea] [Consulta: 15 de Marzo de 2022] Disponible en: <https://ascelibrary.org/doi/10.1061/%28ASCE%29AS.1943-5525.0001331>

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Trabajo de Fin de Grado

Libro 2: Pliego de Condiciones

**Diseño y simulación del sistema de control de apogeo de un cohete sonda.
Proyecto Faraday Rocketry UPV.**



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Autor: Antoni González Chacón

Tutor: Sergio García-Nieto Rodríguez

Grado en Ingeniería Aeroespacial

Curso 2021-2022

Índice

Índice	I
2.1. INTRODUCCIÓN	1
2.2. DESCRIPCIÓN DE LOS TRABAJOS	1
2.3. CONDICIONES DE EJECUCIÓN	1
2.4. CONDICIONES DE MATERIALES	3

2.1. Introducción

El pliego de condiciones es un documento donde se especifican las condiciones de los materiales y de ejecución para que pueda llevarse a cabo el proyecto según esas especificaciones, con el objetivo de aclarar los requisitos y evitar malinterpretaciones.

Primeramente, se van a explicar cada uno de los trabajos que se llevan a cabo para luego pasar a comentar las condiciones que se aplican en cada uno de ellos

2.2. Descripción de los trabajos

En este apartado se explican cada uno de los trabajos principales que se tienen que realizar para entender las condiciones que se van a aplicar a cada uno de ellos en los siguientes apartados.

- Estudio del Modelo no Lineal: Este trabajo incluye todas las tareas derivadas de la obtención del modelo no lineal, ya sea la búsqueda de información para obtener las ecuaciones generales como el estudio de las consideraciones tomadas y las particularizaciones para el modelo de estudio.
- Implementación en Matlab Simulink: Este trabajo incluye todo el montaje de las ecuaciones obtenidas en el trabajo anterior en el lienzo de Simulink para su posterior simulación y ajuste de los controladores necesarios.
- Ajuste de los sistemas de control: Este trabajo incluye todas las tareas de control que se han realizado en el proyecto, desde la obtención de la función de transferencia hasta el ajuste del controlador final.

2.3. Condiciones de ejecución

Se van a explicar las condiciones de ejecución de cada uno de los trabajos que se van a realizar. Además, cada uno de ellos se va a desglosar en sus tareas más importantes por orden de ejecución.

- **Estudio del Modelo no Lineal**

1. Desarrollo de las consideraciones previas:

- Definir las consideraciones a tomar.
- Comprobar la validez de las hipótesis tomadas.
- Definir los ejes de referencia y de los ángulos de Euler.

2. Desarrollo de las ecuaciones del movimiento:

- Obtener la ecuación que rige cada uno de los parámetros que influyen en el equilibrio de fuerzas y en el equilibrio de momentos en el sistema de referencia basado en el cuerpo.

3. Definición de parámetros:

- Recopilar los componentes que componen el cohete que influyen en el movimiento del mismo.
- Obtener la posición del CDG mediante OpenRocket.
- Obtener el tensor de inercia mediante el modelo CAD del cohete completo.
- Desarrollar las ecuaciones de variación de los parámetros dependientes del cambio de masa mediante interpolación por mínimos cuadrados.
- Validar los datos de Drag proporcionados.

• Implementación en Matlab Simulink

1. Configuración del software:

- Establecer el tiempo de muestreo en 0.001 segundos.

2. Implementación de funciones:

- Utilizar del bloque “Matlab Function” para la implementación de las funciones de los parámetros dependientes del tiempo con el bloque “Clock” como entrada de la misma.
- Utilizar lenguaje de programación Matlab para obtener el valor de salida de cada una de las funciones.
- Implementar de las funciones adicionales de manera explícita separando las entradas vectoriales por cada uno de sus elementos.

3. Implementación de las Ecuaciones del movimiento:

- Utilizar las ecuaciones del movimiento aislando las variables \dot{v} y \dot{w} para su posterior integración.
- Utilizar el bloque “PID Controller” de Simulink para introducir los parámetros del controlador.
- Limitar la acción de los aerofrenos a la etapa de ascenso del cohete mediante el bloque “Switch” de Simulink.
- Comprobar la coherencia de la ecuación del Drag que rige la señal de referencia.

- **Ajuste de los sistemas de control**

1. Obtención de la FdT

- Obtener la señal de muestra mediante la simulación del modelo mediante el establecimiento de el valor de alfa con una función escalón de amplitud 0.5 en el instante 8 segundos.
- Utilizar la aplicación System Identification para la estimación de la FdT.
- Comprobar que el ajuste de la FdT estimada supera el 99%.

2. Ajuste del PID de los estabilizadores

- Utilizar la herramienta pidTuner para establecer los parámetros del PID.
- Ajustar el controlador con el mayor valor de robustez y tiempo de establecimiento que permita la estabilidad del sistema.
- Comprobar que ambos PID utilizados tienen los mismo parámetros para obtener los mismo datos en el desplazamiento lateral en cada uno de los ejes del plano horizontal.

3. Ajuste del PID del control de apogeo

- Priorizar la determinación del tipo de controlador por el criterio de simplicidad del controlador antes de por la utilización del método de cancelación de polos del sistema para evitar desprestigiar el PI.
- Comprobar que el PID ajustado logra error nulo y un movimiento no brusco de los aerofrenos.
- Comprobar que efectivamente se logra la altura deseada.

2.4. Condiciones de materiales

Al tratarse de un proyecto puramente realizado virtualmente, los materiales necesarios para su ejecución son simplemente los programas utilizados durante el transcurso del proyecto.

Se requiere tener instalado en el dispositivo de desarrollo del proyecto los programas Matlab (con los paquetes Simulink, System Identification Toolbox, Simulink Control Design y Control System Design and Analysis), Autodesk Fusion 360, Microsoft Office Excel y OpenRocket, junto con las licencias de utilización de cada uno de ellos para poder utilizarlos.

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Trabajo de Fin de Grado

Libro 3: Presupuestos

**Diseño y simulación del sistema de control de apogeo de un cohete sonda.
Proyecto Faraday Rocketry UPV.**



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Autor: Antoni González Chacón

Tutor: Sergio García-Nieto Rodríguez

Grado en Ingeniería Aeroespacial

Curso 2021-2022

Índice

Índice	I
3.1. INTRODUCCIÓN	1
3.2. HONORARIOS DEL INGENIERO	1
3.3. COSTE DE LOS DISPOSITIVOS	1
3.4. GASTO ENERGÉTICO	3
3.5. COSTE DEL SOFTWARE UTILIZADO	3
3.6. COSTE TOTAL DEL PROYECTO	3
3.7. BIBLIOGRAFÍA	3

3.1. Introducción

En este documento se detallarán los diferentes puntos del proyecto que suponen algún tipo de coste económico.

Dichos puntos son el sueldo que debe cobrar el ingeniero o ingenieros que realicen cada una de las tareas que se precisan para elaborar el proyecto, el coste de los dispositivos utilizados para realizar cada una de las tareas, el coste de la energía utilizada para dar potencia a todos los sistemas que la requieren y el coste de los programas de pago utilizados para el desarrollo del proyecto.

El coste total del proyecto se corresponde con la suma del coste de cada uno de estos puntos por separado.

3.2. Honorarios del ingeniero

Para el estudio económico hay que tener en cuenta primeramente el pago al ingeniero que ha realizado el diseño, teniendo en cuenta los diferentes tipos de tareas realizados, los conocimientos necesarios para llevar cada una a cabo, el tiempo de trabajo y el precio de dicho tiempo.

Para realizar este estudio se han tenido que hacer diferentes tipos de tareas. Una búsqueda de información, una búsqueda o cálculo de todos los parámetros del modelo necesarios y un diseño en Matlab Simulink.

Como los dos primeros son realizados por un ingeniero júnior sin experiencia previa el coste de la hora es de 20€, mientras que para la fase de diseño en Matlab Simulink se necesita un conocimiento y soltura en este software por lo que el precio asciende a 25€/h (GLASSDOOR 2022).

Tarea	Precio/hora (€/h)	Tiempo invertido(h)	Precio total (€)
Búsqueda de información	20	50	1000
Desarrollo de los parámetros del modelo	20	50	1000
Diseño en Matlab Simulink	25	150	3750
Redacción del documento	20	40	800
	SUMA	290	6550
		Total sin impuestos	6550
		I.V.A. (21%)	1375.50
		Total	7925.50

Tabla 1: Desglose de precios de los honorarios del ingeniero

3.3. Coste de los dispositivos

Este trabajo en su totalidad solo ha requerido de un dispositivo, un ordenador portátil de uso personal. En este caso se ha utilizado un Acer Nitro 5 AN515-57 (ACER STORE 2022). Cabe destacar que no se debe contabilizar el precio íntegro del dispositivo sino tan solo el porcentaje utilizado de la vida útil del mismo (TECFYS 2022). En otras palabras, el gasto de hardware que se ha hecho.

Además, en las especificaciones del portátil encontramos la potencia de carga, que nos servirá para estimar el gasto energético del estudio en el siguiente apartado.

Dispositivo	Precio de compra (€)	Vida útil (años)	Tiempo invertido (h)	Porcentaje (%)	Coste total
Acer Nitro 5 AN515-57	1099 €	4	290	0.828%	9.10€

Tabla 2: Desglose de precios de los dispositivos utilizados

3.4. Gasto energético

Además, también se tiene que tener en cuenta el coste de la energía utilizada para realizar el estudio. Este coste simplemente incluye el gasto energético del ordenador (ACER STORE 2022), que asumiremos que ha estado siempre enchufado a la corriente, más el gasto energético de las 3 bombillas (LEROY MERLIN 2022) de la habitación donde se ha realizado el trabajo, teniendo en cuenta el precio de la luz (TARIFALUZAHOA 2022) en el momento de realización de este apartado, teniendo en cuenta que puede haber variado durante las 290 h de utilización.

$$\text{Consumo energético} = (180W + 60W \cdot 3) \cdot 290 h = 104400 Wh = 104.4 kWh$$

Así pues, en la Tabla 3 se resume el coste energético total.

Potencia usada (W)	Consumo total (kWh)	Precio del kWh (€/kWh)	Precio total (€)
360	104.4	0.39375	41.10

Tabla 3: Desglose de precios del gasto energético

3.5. Coste del software utilizado

Como se ha comentado, el trabajo se ha realizado en su mayor medida por el software de pago Matlab y sus paquetes de herramientas como Simulink y System Identification Toolbox entre otros.

El coste de la licencia es de 840€ al año o 2100€ sin límite de tiempo (MATHWORKS 2022). En este caso se va a utilizar el coste de 840€ para poder hacer el cálculo de horas, en este caso:

$$\text{Uso posible} = 1 \text{ año} \cdot 365 \text{ días/año} \cdot 24 \text{ horas/día} = 8760 \text{ horas}$$

Además, se han utilizado minoritariamente los programas de pago Microsoft Office Excel (MICROSOFT, 2022) y Autodesk Fusion 360 (AUTODESK, 2022).

Así pues, en la Tabla 4 se detalla el precio del software:

Software	Licencia	Precio de licencia (€/tiempo)	Uso posible (h)	Uso (h)	Porcentaje (%)	Precio total (€)
Matlab Simulink	Anual	840	8760	150	1.71	14.38
Autodesk Fusion 360	Anual	402	8760	10	0.114	0.46
Microsoft Office Excel	Anual	69	8760	15	0.171	0.12
SUMA						14.96

Tabla 4: Desglose de precios del software

3.6. Coste total del proyecto

Así pues, teniendo en cuenta todos los puntos mencionados anteriormente, se obtiene el coste total del proyecto sumando los costes de cada uno de ellos.

Sección	Coste (€)
Honorarios del ingeniero	7925.50
Coste de los dispositivos	9.10
Gasto energético	41.10
Coste del software utilizado	14.96
GASTO TOTAL	7990.66

Tabla 5: Desglose de precios y coste total del proyecto

3.7. Bibliografía

GLASSDOOR, 2022. *Sueldo para el puesto de Ingeniero Junior en España* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: https://www.glassdoor.es/Sueldos/ingeniero-junior-sueldo-SRCH_KO0,16.htm

ACER STORE, 2022. *Acer Nitro 5 Portátil Gaming | AN515-57 | Negro* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://store.acer.com/es-es/acer-nitro-5-portatil-gaming-an515-57-negro-nh-qeseb-001>



TECFYS, 2022. *¿Conoces la vida media de tu ordenador?* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://tecfys.com/blog/conoces-la-vida-media-de-tu-ordenador/>

LEROY MERLIN, 2022. *¿Cuánto consume una bombilla incandescente?* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://comunidad.leroymerlin.es/t5/Bricopedia-Iluminaci%C3%B3n-y/Cu%C3%A1nto-consume-una-bombilla-incandescente/ta-p/11384#:~:text=Respuesta,40%2C%2060%20y%20100%20wattios.>

TARIFALUZAHORA, 2022. *Precio de la luz por horas* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://tarifaluzhora.es/>

MATHWORKS, 2022. *Pricing and Licencien* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://es.mathworks.com/pricing-licensing.html>

AUTODESK, 2022. *Comprar Fusion 360* [en línea]. Consulta: 9 de Julio 2022]. Disponible en: <https://www.autodesk.es/products/fusion-360/overview?term=1-YEAR&tab=subscription>

MICROSOFT, 2022. *Soluciones para tu ajetreada vida, confirmadas* [en línea]. [Consulta: 9 de Julio 2022]. Disponible en: <https://www.microsoft.com/es-ES/microsoft-365/buy/compare-all-microsoft-365-products>